EXHIBIT AA

ZATARAIN I

OPENING EXPERT REPORT OF ARTHUR ZATATRAIN, PE

FEBRUARY 24, 2012

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

ROCKWELL AUTOMATION, INC. and ROCKWELL AUTOMATION TECHNOLOGIES, INC., Plaintiffs, v. WAGO CORPORATION and WAGO KONTAKTTECHNIK GmbH & CO. KG, Defendants.	Case No. 10-CV-718-WMC

EXPERT REPORT OF ARTHUR ZATARAIN UNDER RULE 26(a)(2)(B)
ON BEHALF OF PLAINTIFFS ROCKWELL AUTOMATION, INC.
AND ROCKWELL AUTOMATION TECHNOLOGIES, INC.

1. I, Arthur Zatarain, have been retained by Plaintiffs Rockwell Automation, Inc. and Rockwell Automation Technologies, Inc. ("Rockwell") as a technical expert consultant and potential testifying expert witness in this lawsuit filed against WAGO Corporation and WAGO Kontakttechnik GmbH & Co. KG. In this report, unless the context requires that I specify a particular Defendant, I will refer to both WAGO Corporation and WAGO Kontakttechnik GmbH & Co. KG ("WAGO Germany") collectively as "WAGO."

2. I am a licensed electrical and control systems engineer with over 30 years experience in technical areas relevant to the patents in dispute. Details of my credentials and experience are included later in this report, with a current CV included as Exhibit A.

3. I understand that I am submitting this report in compliance with the requirements of Rule 26(a)(2)(B) of the Federal Rules of Civil Procedure and the Court's scheduling order. I prepared all technical aspects of this report with assistance on legal aspects, citations, and formatting provided by Plaintiffs' counsel, however my opinions addressed herein are my own.

4. I anticipate my testimony will concern my opinions concerning the asserted patents in this lawsuit, the field of art concerning the asserted patents, the state of the art at the time of filing of the asserted patents, the level of ordinary skill in the art at the time of filing the patents, whether WAGO's products practice one or more of the asserted claims, whether WAGO infringes any claim of the asserted patents, and the information contained in this report if called as a witness. I also may provide a technical tutorial regarding the teachings of the asserted patents and related technologies.

5. Because discovery is ongoing, I reserve the right to amend or supplement this report after receiving additional discovery material, testimony transcripts, expert report, or any other relevant information of which I become aware after I have signed this report.

I. INFORMATION CONSIDERED IN PREPARING THIS REPORT

6. In forming the opinions set forth herein, I relied upon my background, education, work experience, general knowledge and experience in the field of industrial control systems.

7. A list of the documents and materials that I reviewed during my work on this lawsuit and that is related to this report is attached as Exhibit B and listed below. I may cite to these documents for additional support for any of my opinions in deposition or trial, beyond what I expressly cite in this report.

8. Among other things, I have reviewed each of the patents involved in this lawsuit and the records of the United States Patent and Trademark Office related to the applications that led to those patents (which I understand is called the "prosecution history").

9. I also have reviewed transcripts from the depositions of Thomas J. Artmann (as a Rule 30(b)(6) witness), Mark W. DeCramer (as a Rule 30(b)(6) witness), Gregory Rinn (as both

a Rule 30(b)(6) witness and as an individual), and Thomas Albers (as both a Rule 30(b)(6) witness and as an individual).

10. I also have reviewed numerous manuals and other technical information that I understand was either produced by WAGO in this lawsuit or is available on WAGO Corporation's website (*www.wago.us*) or WAGO Germany's website (*www.wago.com*).

11. I received technical support from WAGO Corporation on how to operate the WAGO-I/O System products.

12. In connection with my anticipated testimony in this action, I may use as exhibits various documents produced in this case, including but not limited to the documents I reviewed above. I have not yet selected the particular exhibits that might be used. In addition, I may create or assist in the creation of certain demonstrative exhibits to assist with testimony.

II. SUMMARY OF THE PATENTS AT ISSUE

13. I understand that Plaintiffs are asserting that Defendants WAGO Corporation and WAGO Kontakttechnik GmbH & Co. KG have infringed and are infringing the following claims from the following patents:

a. United States Patent No. 6,745,232: claims 1, 2, 3, 5, 10, 11, and 14;

b. United States Patent No. 7,123,974: claims 1-3, 5, 6, 9, 10, 14, 16, 24, and 29;

c. United States Patent No. 6,801,813: claims 1-7, 10-17, and 20-22; and

d. United States Patent No. 7,065,415: claims 1-5, and 8.

A. U.S. Patent No. 6,745,232

14. The '232 patent is entitled "Strobed Synchronization Providing Diagnostics in a Distributed System." The '232 patent issued on June 1, 2004 from an application filed on August 23, 2000, and it names Steven P. Blech, Gregory A. Majcher, and John P. Caspers as inventors.

15. The '232 patent is directed to a runtime debugging and diagnostics method and system for industrial control systems. The disclosed embodiments allow a user to regulate processing of a control device in a step mode during execution of a stored program. The patent discloses methods of storing an executable program in a programmable logic controller ("PLC"), where the PLC communicates with devices in a networked system to receive instructions to suspend execution, change modes of execution, or run in a step mode. The step mode can be for a specified number of iterations of the code, or for a specified time period. Further, the patent describes the structure of mode change messages or step mode messages exchanged between the executing PLC and a communications medium.

B. U.S. Patent No. 7,123,974

16. The '974 patent is entitled "System and Methodology Providing Audit Recording and Tracking in Real Time Industrial Controller Environment." The '974 patent issued on October 17, 2006 from an application filed on November 19, 2002, and it names Jeffrey L. Hamilton as the inventor.

17. Aspects of the '974 patent are directed to systems and methods for "real time" automated monitoring, recording, and tracking of interactions with an industrial control component. A recording component on the control component (or on a remote device) records certain real-time interactions (*e.g.*, access records; value forcing operations; clearing/setting

counter, register, or status area of memory). A tracking component aggregates these interactions. The aggregated interactions can be used to generate audit data related to the control component.

C. U.S. Patent No. 6,801,813 and U.S. Patent No. 7,065,415

18. The '813 patent is entitled "Method for Consistent Storage of Data in an Industrial Controller." The '813 patent issued on October 5, 2004 from an application filed on July 30, 2001, and it names James J. Kay, David Allen Johnston, Shelly Lynn Urdaneta, and Stuart Blair Siegel as inventors.

19. The '415 patent is entitled "Method for Consistent Storage of Data in an Industrial Controller." The '415 patent issued on June 20, 2006 from an application filed on August 23, 2004 (which was based on the application that led to the '813 patent that was filed on July 30, 2001), and it names James J. Kay, David Allen Johnston, Shelly Lynn Urdaneta, and Stuart Blair Siegel as inventors.

20. The '813 and '415 patents are directed to systems and methods of employing a file system in the memory of a PLC. An editor allows a user to write an industrial control program that utilizes the file system services on the PLC. An execution engine operable in the PLC is adapted to interpret instructions that invoke the services of the file system on the PLC. The file system on the PLC can be used to log and retrieve measured data and trend data. The file system can be used to load "recipe files" from local or remote storage location. The execution engine and the file system allow loading user-defined routine files at run-time from a file storage location. Typical file service storage locations could be a memory device residing within (or attached to) a PLC, on a local server, or at a remote location.

III. OVERVIEW OF THE TECHNOLOGY AT ISSUE

21. The general subject matter of the patents in this lawsuit relates to industrial control systems using PLCs. These systems are used to monitor and control industrial processes, manufacturing equipment, and other factory automation both directly and in distributed (i.e. networked) configurations.

22. At the core of the industrial control system is a logic processor, typically termed Programmable Logic Controller, or a PLC. These are computer-based devices with underlying technology based on a wide range of small and large computer platforms. A PLC uses a control program to measure one or more physical process inputs or other data variables reflecting the status of a controlled system. The PLC then uses the control program to alter process outputs or variables as required by the system being controlled. The inputs and outputs may be binary values, (e.g. simply on or off), or they can be numeric or analog values residing in a continuous range (e.g. 0 to 100% of a measurement scale).

23. The physically measured inputs received into such systems, and the outputs transmitted by the systems, generally pass through one or more input/output (I/O) modules. These I/O modules serve as a physical interface to the PLC. The I/O may be connected directly to the PLC processor, or it may be located remotely and connected via a communications network.

24. PLC's function according to user-designed logic programs. Some PLC systems allow program development directly on the PLC itself. A more common method of creating a control program involves using an access tool containing a development editor residing on a separate device, such as a personal computer. Using the access tool, the programmer designs data structures and writes logic instructions to create a complete control program that will

eventually operate on the target PLC. Once the control program is complete, the access tool processes the programmed control instructions into a format compatible with the target PLC's logic processor. The completely processed logic program is then downloaded from the personal computer to the target PLC where it is stored in memory and eventually executed to perform one or more process control tasks.

25. Some PLCs allow the user to make a limited amount of "online" changes to the logic program by connection of the editor tool to the PLC. With this method, changes are made directly to the program and data already stored on the PLC. The online method is alternate to modifying the program on the editor tool and then downloading an entirely new processed program.

26. Because the PLC logic program is either pre-processed on a personal computer using the access tool, or modified directly on the PLC with the access tool, no logic changes can be made without use of an access tool connected to the PLC.

A. '813 and '415 Patents

27. PLCs typically have internal memory storage devices and data formats that are not readily accessible to external computer systems. Data storage on the PLC is also typically very limited as compared to other computer systems. Data exchange between a PLC and other computers requires extraction of the data in the native PLC format for conversion to formats useful outside of the PLC. This extraction is typically done with a communications network between the PLC and an external system that performs the data conversion.

28. Plaintiffs' '415 and '813 patents disclose the use of an onboard file system to load and save data directly by the PLC. This includes data exchange using removable storage media such as a USB drive or Compact Flash card. This data could include the formats previously used

in the PLC's internal storage as well as additional data formats, such as comma separated (CSV) files, that were not available to PLC's that lacked a file system.

29. Plaintiffs' '813 patent also provides for the use of an onboard PLC file system to load user routines from the PLC memory based on header instructions in the PLC control program.

30. The use of a standard file system allows direct exchange of PLC data with other computers using a compatible file system. The PLC file system increases PLC flexibility by incorporating features previously available only on personal computers and similar devices. The ability to load user routines directly on the PLC provides, among other things, the ability to alter the control logic using the PLC file system to load different user routines without use of an access tool.

B. '232 Patent

31. Logic programs running within a PLC normally operate in a continuous cyclic manner, with cycle times measured in milliseconds. Alternately, some portions of a PLC program function only under certain circumstances, such as when a particular process condition occurs. Examination of the fast-moving or sporadic program steps and related data variables requires attachment of an online access tool. However, examination of an operating program is especially difficult when the PLC logic is experiencing continuous and rapid changes in I/O values or program logic states.

32. The Plaintiffs' '232 patent discloses an improved way of examining PLC program logic and data on an operating PLC. The disclosure provides for the running PLC program to suspend operation at one or more designated locations. The suspended operation occurs when the PLC receives a message from a communications medium. Using an online access tool, this

ability provides the user with an "inside look" at specific program functions when they first occur, and also when they occur in subsequent cycles. While suspended, the PLC can provide data to the communications medium to allow examination and modification of the data associated with the suspended state. Once suspended, execution of the program can proceed in a stepped, timed, or continuous mode as directed by further received messages.

33. The ability to halt a program and then step through its sequential functions simplifies logic error detection and correction. The ability to view and modify data related to the PLC's operation while the program is suspended simplifies debugging of control programs by giving the user direct access to the actual process data being used by the actual PLC running or suspended program.

C. '974 Patent

34. Industrial control systems are used for manufacturing, process control, and safety. Some of these applications require extensive documentation of the overall control process in order to manage and verify the quality of products that are produced. The patent discloses that one such requirement relates to FDA regulation "21 CFR Part 11" that outlines an electronic validation procedure to document changes to important production processes. Such requirements include, for example, how records are kept, who can access such records, and what procedures/controls are designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records.

35. Given the programmable nature of PLC systems, intentional changes in operating conditions can be easily made to various aspects of the control system. Such changes to the PLC may occur during startup conditions, maintenance activities, or special operations. Even the smallest change to PLC or process data can place an immediate and significant consequence on

the system even if the underlying PLC program remains unchanged. For example, input and output points can be intentionally "forced" to specific values by an operator manipulating the system for a special purpose. Other changes such as clearing/setting a counter, register, and/or status of the PLC memory can also produce dramatic, process-changing results. Without proper management, recording of these online changes may not occur as required. Plaintiffs' '974 patent addresses this problem with an invention that automatically records and tracks relevant interactions with an industrial control system such as a PLC.

36. Plaintiffs' '974 patent discloses using a recording component that logs real time interaction with one or more industrial control components. It further discloses a tracking component to aggregate the real time interactions to facilitate generation of audit data relating to the control components. These components can be associated with an access tool that interacts with the control component over a network.

37. The ability to record and track control system interactions in real time represents an improved "management of change" process for industrial controllers. The '974 patent provides for automatic and secure record keeping to assist the entity responsible for meeting any regulatory or other requirement for monitoring and tracking changes to the industrial control process.

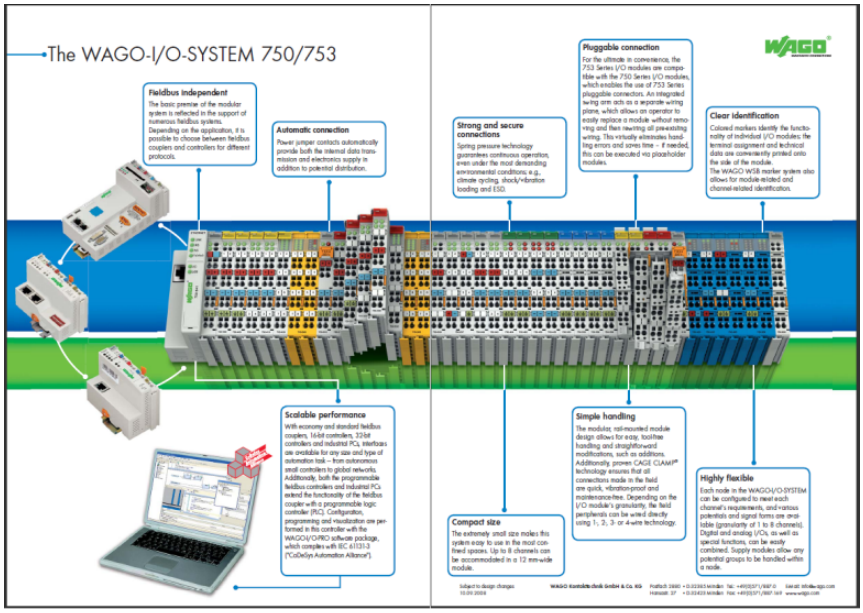
IV. DEFENDANTS' PRODUCTS

38. I have been advised by Plaintiff's counsel that Defendants make, use, offer for sale, sell, and import into the United States various devices under the name "WAGO-I/O-System." (*See, e.g.*, http://www.wago.us/products/276.htm; http://www.wago.com/wagoweb/documentation/index_e.htm;

http://www.wagocatalog.com/okv3/index.asp?cid=51&lid=5&str_from_home=first; see also WAGO I/O SYSTEM - Full Line Catalog 2010/2011).

39. I have also been advised that WAGO makes, offers to sell, sells, and imports a collection of industrial control products that can be installed in many different configurations. Many of these products can be mounted on a standardized "DIN rail" within control cabinets, or on or near equipment on a factory floor, generally in an industrial or commercial environment. For installation, many of the products slide together on the DIN rail to make power and data connections with each other. The first module, typically located on the left, is always a controller or bus coupler that manages the complete control system "node." The WAGO controllers are programmable for both communications functions as well as control functions.

40. To accomplish the input/output functions, customers connect many different types of I/O modules to the controller or bus coupler to accommodate their particular process control needs. Finally, an end module completes the lineup of I/O modules at the end of the node. As shown below, the completed control node resembles slices of bread aligned to form a complete "loaf":



(http://www.wago.com/wagoweb/documentation/750/eng_dat/d0750xxxx_xxxxxxxx_0en.pdf).

41. The above photo shows a WAGO Series 750-841 Ethernet controller connected to many other WAGO I/O System modules. This photo also shows that WAGO offers different fieldbus couplers and controllers (three of which are shown on the left) that can be used. WAGO's customer's choice of controllers, couplers, and other modules depends on the needs of their control system. This photo also shows that WAGO's programmable devices can be programmed using WAGO-I/O-PRO software package, which it refers to as WAGO-I/O-PRO CAA or CoDeSys.

42. One WAGO manual explains the overall construction of the WAGO I/O products as follows:

> The coupler/controller contains the field bus interface, electronics and a power supply terminal. The field bus interface forms the physical interface to the relevant field bus. The electronics process the data of the bus modules and make it available for the field bus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal. The field bus coupler communicates via the relevant field bus. The programmable field bus controller (PFC) enables the implementation of additional PLC functions. Programming is done with the **WAGO-I/O-*PRO*** in accordance with IEC 61131-3.

(WAGO I/O System 758, Modular I/O System, WAGO-I/O-IPC, 758-870/xxx-xxx, Manual, Technical description, installation and configuration, Version 1.1.0" (2004), § 2.1, p. 15 (available at http://www.wago.com/wagoweb/documentation/758/eng_manu/870/m287000e.pdf).

43. WAGO sells many programmable controllers with many different product numbers and names such as:

- "758-xxx," "WAGO-I/O-IPC (SYSTEM 758)," "System 758," "I/O-IPC," and "IPC"
- "750-841," "750-842," and "750-843," each of which WAGO describes as an "ETHERNET TCP/IP Programmable Fieldbus Controller"
- "750-871," which WAGO describes as an "ETHERNET TCP/IP 2 Port Programmable Fieldbus Controller"
- "750-872," which WAGO describes as a "Programmable Fieldbus Controller for Telecontrol Applications"
- "750-873," which WAGO describes as an "ETHERNET TCP/IP & RS232 Programmable Fieldbus Controller"
- "750-880" and "750-881," each of which WAGO describes as an "ETHERNET Programmable Fieldbus Controller"
- "750-882," which WAGO describes as an "ETHERNET Programmable Media-Redundancy Fieldbus Controller"
- "750-833," which WAGO describes as a "PROFIBUS DP/V1 Programmable Fieldbus Controller"
- "750-804," which WAGO describes as an "INTERBUS Programmable Fieldbus Controller"
- "750-806," which WAGO describes as a "DeviceNet Programmable Fieldbus Controller"
- "750-837" and "750-838," each of which WAGO describes as a "CANopen Programmable Fieldbus Controller"

- "750-812," "750-814," "750-815," and "750-816," each of which WAGO describes as a MODBUS Programmable Fieldbus Controller
- "750-819," which WAGO describes as a "LonWorks® Programmable Fieldbus Controller"
- "750-830," which WAGO describes as a "BACnet/IP Programmable Fieldbus Controller"
- "750-849," which WAGO describes as a "KNX IP Programmable Fieldbus Controller"
- "750-860," which WAGO describes as a "Linux® Programmable Fieldbus Controller"
- "750-863," which WAGO describes as a "Linux® Programmable Fieldbus Controller, RS-232"

(*See* http://www.wago.com/wagoweb/documentation/index_e.htm, "Technical Documentation," "Automation," "Data Sheets, Manuals, Other Information," "Fieldbus Coupler and Fieldbus Programmable Controller").

44. Based on my review of the documents cited in this report, including but not limited to deposition testimony, it is my understanding that each of the accused programmable controllers generally perform the same I/O and interface functions using essentially the same programmable features. The major differences among them relate to 1) the communication protocols with which they operate/communicate, and 2) the controller's computer speed and program memory, and 3) the availability of computer-related "ports" for displays, keyboards, communications, and removable memory devices.

45. WAGO sells many bus couplers with many different product numbers and names such as:

- "750-341" and "750-342," each of which WAGO describes as an "ETHERNET TCP/IP Fieldbus Coupler"
- "750-352," which WAGO describes as an "ETHERNET Fieldbus Coupler"
- "750-352/020-000," which WAGO describes as an "ETHERNET TCP/IP ECO Fieldbus Coupler"
- "750-340" and "750-370," each of which WAGO describes as a "PROFINET IO Fieldbus Coupler"
- "750-350" which WAGO describes as an "ETHERNET Powerlink Fieldbus Coupler"
- "750-301" and "750-303," each of which WAGO describes as a "PROFIBUS DP/FMS Fieldbus Coupler"
- "750-333," which WAGO describes as a "PROFIBUS DP/V1 Fieldbus Coupler"
- "750-323," "750-331," and "343," each of which WAGO describes as a "PROFIBUS DP Fieldbus Coupler"
- "750-304," "750-324," "750-334," "750-344," and "750-345," each of which WAGO describes as an "INTERBUS Fieldbus Coupler"
- "750-306" and "750-346," each of which WAGO describes as a "DeviceNet Fieldbus Coupler"

- "750-307," "750-337," "750-338," "750-347," and "750-348," each of which WAGO describes as a "CANopen Fieldbus Coupler"
- "750-305," which WAGO describes as a "CAL Fieldbus Coupler"
- "750-312," "750-314," "750-315," and "750-316," each of which WAGO describes as a "MODBUS Fieldbus Coupler"
- "750-310," which WAGO describes as a "CC-Link Fieldbus Coupler"
- "750-319," which WAGO describes as a "LonWorks® Fieldbus Coupler"
- "750-319/004-000," which WAGO describes as a "Data Exchange Coupler LON (Peer-to-Peer)"
- "750-300" and "750-320," each of which WAGO describes as an "II/O-LIGHTBUS Fieldbus Coupler"
- "750-313," which WAGO describes as an "SDS Fieldbus Coupler"

46. As one of WAGO's manuals explains, the couplers and controllers of the modular I/O System 750 receive digital and analog signals from the I/O modules and sensors and transmit them to the actuators or higher level control systems. ("WAGO I/O System 758, Modular I/O System, WAGO-I/O-IPC, 758-870/xxx-xxx, Manual, Technical description, installation and configuration, Version 1.1.0" (2004), § 1.1.3, p. 9 (available at http://www.wago.com/wagoweb/documentation/758/eng_manu/870/m287000e.pdf).

47. WAGO documentation states that there are over 400 I/O modules available to use with the WAGO controllers. (See WCP 22838, 22855).

48. As one of WAGO's manuals explains: "You can connect all available I/O modules of the WAGO-I/O-SYSTEM 750/753 to the I/O-IPC. This allows any analog and digital signal from the automation environment to be internally processed or to be made available

to other devices through one of the available interfaces." (WAGO 758-870/000-110 Manual, version 2.0.0 (2009), p. 14).

49. This same WAGO manual describes how WAGO's programmable devices can be programmed: "Automation tasks can be executed in all IEC 61131-3-compatible languages with the programming system CoDeSys 2.3 (WAGO-I/O-PRO CAA). The implementation of the CoDeSys task processing is optimized with real-time extensions in order to provide maximal performance for automation tasks. For visualization, CoDeSys target visualization and web visualization are also available in addition to the development environment." (WAGO 758-870/000-110 Manual, version 2.0.0 (2009), p. 14).

50. WAGO's website, *www.wago.us.com*, describes the products that it makes, sells, offers to sell, and imports into the United States under the heading "Components for Automation." (http://www.wago.us/products/276.htm). This WAGO website has links for, among other things, the following products: "Ethernet 2.0," "WAGO-I/O-SYSTEM 750," "WAGO-I/O-SYSTEM 753," "WAGO-I/O-SYSTEM 755," "WAGO-I/O-SYSTEM 757," "WAGO-I/O-IPC (SYSTEM 758)," "I/O Software (Series 759)," "WAGO SPEEDWAY 767," "PERSPECTO – Monitors and Panels," "Condition Monitoring," "Interface Modules," "Overvoltage Protection," "Power Supplies," and "JUMPFLEX Signal Conditioners and Relays."

51. WAGO's website describes its "Ethernet 2.0" products as follows: "For automation engineers who strive to enhance machine or process performance, speed system development and reduce automation costs, the WAGO Ethernet 2.0 programmable controllers provide game-changing flexibility and a proven architecture that simply works."

(http://www.wago.us/products/276.htm). WAGO's website describes its "Ethernet 2.0" products' "Extended Features" as follows:

- IT protocols: HTTP, BootP, DHCP, DNS, SNTP*, SNMP*, FTP
- Industrial networks: Modbus/TCP and EtherNet/IP* built-in
- Minimum of 1MB program memory and 512kb data memory
- Web-based management and visualization
- 2MB for built-in file system and web server functions
- Access to over 400 digital, analog and special function I/O modules
- 24-month warranty, free technical support, and free factory training

*750-881, 750-880, 750-880/025-000

(http://www.wago.us/products/37712.htm).

52. WAGO's website describes its "WAGO-I/O-SYSTEM 750" products as follows: "The WAGO-I/O-SYSTEM 750 is the most compact, modular and fieldbus-independent I/O system for decentralized automation." (http://www.wago.us/products/276.htm). WAGO's website states the following about these products: "The basic premise of the modular system is reflected in the support of numerous fieldbus systems. Depending on the application, it is possible to choose between fieldbus couplers and controllers for different protocols." (http://www.wago.us/products/337.htm).

53. One of WAGO's manuals describes the system as follows: "The WAGO-I/O-SYSTEM 750 is a modular, field bus independent I/O system. It is comprised of a field bus coupler/controller (1) and connected field bus modules (2) for any type of signal. Together, these make up the field bus node. The end module (3) completes the node." ("WAGO I/O System 758, Modular I/O System, WAGO-I/O-IPC, 758-870/xxx-xxx, Manual, Technical description, installation and configuration, Version 1.1.0" (2004), § 2.1, p. 15).

54. WAGO's website provides more detailed descriptions of its "WAGO-I/O-SYSTEM 750" products under the following headings and with the following descriptions:

a. Components – "The WAGO-I/O-SYSTEM 750 makes it possible: complete in-the-field wiring of the I/O modules, without definition of the final fieldbus. Combination of digital / analog inputs and outputs with different potentials, capacity and signals on one fieldbus node." (http://www.wago.us/products/337.htm). The WAGO website has further information about these types of products under the headings: "fielbus couplers, controllers;" "I/O terminals;" "I/O special terminals;" "PROFIsafe safety modules;" "EEx i bus terminals;" "System terminals;" and "Fieldbus plugs." (http://www.wago.us/products/1855.htm).

b. Function Modules – "Versatile and flexible - more than 300 different 1, 2, 4, 8, and 16-channel function modules are available." (http://www.wago.us/products/337.htm). The WAGO website has further information about these types of products under the headings: "Digital input modules" (2-Channel Digital Input Modules, 2-channel digital special function, 4-Channel Digital Input Modules, 8-Channel Digital Input Modules, 16-Channel Digital Input Modules, and Safety Modules (ProfiSafe)); "Digital Output Modules" (1-Channel Digital Output Modules, 2-Channel Digital Output Modules, 2-channel digital special function, 4-Channel Digital Output Modules, 8-Channel Digital Output Modules, 16-Channel Digital Output Modules, Safety Modules (ProfiSafe), and 2-Channel Relay Output Modules); "Analog Input Modules" (1-Channel Analog Input Modules, 2-Channel Analog Input Modules, 4-Channel Analog Input Modules, and Analog Special

Function); "Analog Output Modules" (2-Channel Analog Output Modules, 4-Channel Analog Output Modules, and Analog Special Function); "Drive Technology" (Counter, Distance and Angle Measurement, Positioning, and Vibration Monitoring); "Communication and Gateway" (AS-Interface Master, Radio Inferface, Serial Interface, Module Bus Extension, and Bluetooth RF Transceiver); "Building Services" (1-Channel Digital Output Module, DALI / DSI Master, EnOcean Radio Receiver, MP-Bus (Multi Point Bus), KNX/EIB/TP1 Module, and RTC Module); "System Modules" (Filter module, Field side connection modules, Separation modules, Supply module, and Module bus extension); and "Ex i Intrinsically Safe Modules" (1-Channel Digital Input Module, 2-Channel Digital Input Module, 2-channel digital output module, 2-Channel Digital Input Module, 2-channel analog input module, and 2-channel analog output module). (http://www.wago.us/products/12537.htm).

c. WAGO-I/O-SYSTEM - Ideal Fieldbus Node – "Universal, Compact, Economic - The Ideal Fieldbus Node." (http://www.wago.us/products/337.htm). WAGO's website provides the following description of this product: "With economy and standard fieldbus couplers, 16-bit controllers, 32-bit controllers and industrial PCs, interfaces are available for any size and type of automation task — from autonomous small controllers to global networks. Additionally, both the programmable fieldbus controllers and industrial PCs extend the functionality of the fieldbus coupler with a programmable logic controller

(PLC). Configuration, programming and visualization are performed in this controller with the WAGO-I/O-PRO software package, which complies with IEC 61131-3 ('CoDeSys Automation Alliance')." (http://www.wago.us/products/12524.htm).

d. Specialty Modules – "Whether DALI/DSI Master, stepper controller, radio receiver... - WAGO offers a large variety of specialty modules." (http://www.wago.us/products/337.htm). WAGO's website provides the following description of these products: "Since its market introduction in 1995, the WAGO-I/O-SYSTEM 750 has been continually expanded and complemented. Therefore, many specialty modules are now available that range from interface modules, such as RS-232 C and RS-485, to specialty modules like DALI/DSI Master, radio receiver I/O module, MP-Bus Master, RTC module, KNX/EIB/TP1 modules and Bluetooth® I/O modules." (http://www.wago.us/products/23730.htm). WAGO's website then has additional information under the following headings: "I/O Radio Receiver," "I/O DALI DSI Master Module," "Stepper controller system family," and "Bluetooth RF Transceiver." (http://www.wago.us/products/23730.htm).

e. Highlights – "Since its market introduction in 1995, the WAGO-I/O-SYSTEM 750 has morphed into one of the most important modular fieldbus systems for industry, production and process automation." (http://www.wago.us/products/337.htm). Under this heading, WAGO's website provides additional information about "Linux controller and

WAGO-I/O-IPC" (http://www.wago.us/products/1881.htm), specifically: "Wago is going with Linux ... Linux offers more reliability and investment security for automation than a number of other commercial operating systems. This is why WAGO has been building on this open-source operating system for its Linux controllers and industrial PCs since 2004." (http://www.wago.us/products/19840.htm).

f. Starter Kit – "You too may want to start with the innovative WAGO-I/O-SYSTEM and realize the most up-to-date automation technology in a short time." (http://www.wago.us/products/337.htm). WAGO's website provides links to various "starter kits" that WAGO offers for sale. (http://www.wago.us/products/1926.htm).

g. Applications – "Approved for industrial automation, building automation, marine automation, and on- and offshore applications, the WAGO-I/O-System provides for many different applications - even under aggravated ambient conditions." (http://www.wago.us/products/337.htm). WAGO's website provides several descriptions of how their products have been used. (http://www.wago.us/products/1911.htm).

h. Safe I/O for PROFIBUS and PROFINET – "With the WAGO-I/O-SYSTEM 750/753. Functional and flexible safety." (http://www.wago.us/products/337.htm). WAGO's website provides a description of the "PROFIsafe safety modules" that WAGO offers for sale. (http://www.wago.us/products/1899.htm).

55. WAGO's website describes its "WAGO-I/O-SYSTEM 753" products as follows: "Pluggable I/O Modules: For certain applications, a pre-wiring of components, independent of the electronics, is appropriate." (http://www.wago.us/products/276.htm). WAGO's website provides the following additional description of these products: "Some applications, especially in safety related fields, require a module in the fieldbus node to be exchanged without disturbing the existing wiring. For this purpose, WAGO has added I/O modules with pluggable field wiring to the WAGO-I/O-SYSTEM 750. A pre-wired connector that plugs into the I/O module fulfills this function. The new modules are only slightly higher, the width and the depth remain the same as the existing modules." (http://www.wago.us/products/357.htm).

56. WAGO offers many of the same modules in the 753 Series as are offered in the 750 Series, but there is not a 100% one-to-one correspondence; that is, not all 750 Series modules are available in an equivalent 753 Series version and not all 753 modules exist as 750 modules. (DeCramer 30(b)(6) Dep. Tr. 55:13-19).

57. WAGO's marketing materials state that "WAGO's 753 Series of I/O modules adds the benefit of pluggable field wiring to the WAGO-IO-SYSTEM... This new series of modules is 100% compatible with WAGO 750-Series I/O modules." (See WCP 22689).

58. WAGO's website describes its "WAGO-I/O-SYSTEM 755" products as follows: "Consequently executed decentralized automation ends closely at the machine - frequently so close that there is no room for a protective cabinet or enclosure. WAGO-I/O-SYSTEM 755 was designed for those conditions." (http://www.wago.us/products/276.htm). WAGO's website lists the following applications for these products: severe environmental conditions, standardized connections, installation close to the machine, digital signals, and higher concentration of signals. (http://www.wago.us/products/355.htm).

59. WAGO described the purpose of the 755 system as "a coupling system that can be used in a wet environment or in a non-clean environment." (Albers 30(b)(6) Dep. Tr. 19:4-10).

60. Wago's 758 and 750 controllers act to control the components of the 750, 753 and 755 systems. (See Albers 30(b)(6) Dep. Tr. 20:19-23).

61. WAGO's website describes its "WAGO-I/O-SYSTEM 757" products as follows: "The IP67 Fieldbus system does not offer in every situation the flexibility and cost-effectiveness which would allow the user to achieve a consistent change-over from the IP20 to the IP67 automation system. In such cases, Sensor/actuator boxes are a real alternative." (http://www.wago.us/products/276.htm). WAGO's website provides the following explanation about how this product can be used: "Using the new IP67 sensor/actuator boxes from WAGO, signal acquisition is achieved close to the machine. Through the connection to the WAGO-I/O-SYSTEM 750 (protection type IP20) installed in the switch cabinet, the user obtains maximum performance capability, flexibility and convenience (Fieldbus system, scalability, programmability, etc.)." (http://www.wago.us/products/353.htm).

62. WAGO's website describes its "WAGO-I/O-IPC (SYSTEM 758)" products as follows: "**WAGO-I/O-IPC, compact industrial PC.** The IPC is a cost-effective industrial compact PC solution for real time control applications. Despite its compact design, the IPC integrates all standard PC functions, including network and fieldbus interfaces, for local and remote control applications." (http://www.wago.us/products/276.htm). WAGO's website also describes this product as follows: "With the I/O-IPC, WAGO presents a robust automation device based on Linux 2.6 with real-time expansion. The industry hardened DIN rail-mountable PC offers high computing power in the smallest spaces possible." (http://www.wago.us/products/351.htm). WAGO's website also describes this product as

follows: "The small housing is mountable on DIN-Rail (35 mm). WAGO I/O modules can be directly connected. Typical markets for IPC applications are the high end machinery industry (e.g., packaging, bottling, textiles, production and metal & wood processing). "

(http://www.wago.us/products/351.htm).

63. One of WAGO's manuals describes WAGO's I/O-IPC as follows:

> The I/O-IPC is an industrial PC that combines the functionalities of a PC and a programmable logic control (PLC). I/O modules and the PROFIBUS or CANopen fieldbus can be connected. The fieldbus connections depend on the version of the IPC. All versions have two 10/100BASE-TX Ethernet interfaces.
>
> If using the I/O-IPC as a PLC, it is possible to control all or some of the I/O modules locally by using WAGO-I/O-*PRO* CAA (CoDeSys). WAGO-I/O*PRO* CAA is an IEC 61131-3 compliant programming tool for programming and configuring the I/O-IPC.
>
> The protocols HTTP, BootP, DHCP, SNTP, FTP and SMTP can be used for system management and system diagnostics.
>
> In addition to Ethernet (TCP/UDP) the version-dependent fieldbus protocols PROFIBUS and CANopen are also supported.
>
> Additional library functions extend the versatility of programming functions. The IEC 61131-3 library SysLibRTC.lib, for example, allows the integration of a battery-backed real-time clock with date (resolution 1 second), alarm function and timer. If the power supply is interrupted, the clock will be powered by a battery.
>
> The I/O-IPC is based on a Pentium compatible MMX CPU and is a multitasking tool; i.e., several programs can run simultaneously. The I/O-IPC has an internal server for web-based applications. As standard, the HTML pages in the I/O-IPC contain information about the configuration and the status of the I/O-IPC. A web browser will open and display the HTML pages. Furthermore, it provides a file system that saves individual HTML pages in the I/O-IPC via FTP download.

("WAGO I/O System 758, Modular I/O System, WAGO-I/O-IPC, 758-870/xxx-xxx, Manual, Technical description, installation and configuration, Version 1.1.0" (2004), § 3.1.1, p. 49

(available at

http://www.wago.com/wagoweb/documentation/758/eng_manu/870/m287000e.pdf).

64. A later WAGO manual for the Series 758 product provides the following description:

> The I/O-IPC automation device is a PC that is suitable for use in an industrial environment and is especially distinguished by its different interfaces.
>
> You can connect all available I/O modules of the WAGO-I/O-SYSTEM 750/753 to the I/O-IPC. This allows any analog and digital signals from the automation environment to be internally processed or to be made available to other devices through one of the available interfaces.
>
> Transfer rates of 10 Mbit/s or 100 Mbit/s are possible via two independent Ethernet interfaces in both half-duplex and full-duplex operation.
>
> The firmware installed at delivery is based on Linux with special real-time extension of the RT-Preempt patch. In addition, various user programs are already installed on the I/O-IPC. These include, for example, a Telnet and FTP server, an NTP client, a BootP and DHCP daemon, the CoDeSys runtime environment and other different help programs. For data exchange, there are implemented MODBUS TCP, UDP and RTU servers available. Web-based management using an Internet browser is used for the configuration of user programs.
>
> Automation tasks can be executed in all IEC 61131-3-compatible languages with the programming system CoDeSys 2.3 (WAGO-I/O-PRO CAA). The implementation of the CoDeSys task processing is optimized with real-time extensions in order to provide maximal performance for automation tasks. For visualization, CoDeSys target visualization and web visualization are also available in addition to the development environment.
>
> The I/O-IPC provides a 128 MB program and data memory and a 128kB non-volatile memory. Both clients and servers for TCP or UDP can be programmed via function blocks.

("WAGO 758-870/000-110 Manual, version 2.0.0" (2009), p. 14).

65. WAGO states that both WAGO and Kontron Modular Computers manufacture hardware and software for the IPC products. (See Albers 30(b)(6) Dep. Tr. 58:15-60:7).

66. WAGO states that when a customer takes an IPC out of the box, the Linux operating system is already loaded on the IPC processor. (See Albers 30(b)(6) Dep. Tr. 21:15-18).

67. Some of the WAGO 750 PLCs use the Nucleeus operating system (i.e. 750-841). (See Albers 30(b)(6) Dep. Tr.).

68. WAGO states that the CoDeSys version 2.3 product is used to program the WAGO I/O 758 series of products. (See Albers 30(b)(6) Dep. 47:17-20; see 758 datasheets).

69. WAGO's website describes its "I/O Software (Series 759)" products as follows: "**Software.** Maintenance/programming software and WAGO-TOPLON® for building automation." (http://www.wago.us/products/276.htm). WAGO's website describes the following software that is offered for sale by WAGO:

i. "WAGO-I/O-CHECK 2" – "WAGO-I/O-CHECK 2 is a Windows application for the checking of inputs and outputs and the display of a WAGO-I/O-SYSTEM 750 node. The software is easy to handle and it is not necessary to connect the node to a fieldbus system."

j. "WAGO-I/O-PRO" – "WAGO-I/O-PRO is a programming and visualization tool, allowing users to develop PLC applications for the Programmable Fieldbus Controller of the WAGO-I/OSYSTEM 750 Series."

k. "WAGO Interface Software" – "The OPC standard defines an open industrial interface which can be used by PC-based software components

to transfer data. The interface is based on the OLE (Object Linking and Embedding), COM Component Object Model), and DCOM (Distributed COM) Windows technologies."

l. "Plug-In TOPLON® PRIO" – "With TOPLON PRIO, you can realize the node integration to the LON® network. And forget about incompatibilities between devices by different manufacturers."

m. "Plug-In TOPLON® IF" – "Standard building automation made easy with TOPLON IF. Using the standard LONWORKS® coupler with the LNS Plug-In TOPLON IF, enables you to realize customer requirements quick and efficiently, even without programming knowledge."

n. "Automation Cockpit" – "Integrated development environment for the configuration, planning, and programming of control panels, couplers, and the WAGO-I/O-SYSTEM 750 controllers."

(http://www.wago.us/products/349.htm).

70. I am told that WAGO produced in discovery a CoDeSys brochure that states that the architecture of CoDeSys can be divided in three basic layers: (1) The Development Layer which contains the PLC programming system CoDeSys with the complete online and offline functionality, the compilers as well as additional components for configuration, visualization etc.; (2) Communication Layer; and (3) Device Layer which states "[b]efore a device can be operated with the IEC 61131-3 development tool CoDeSys, its counterpart on the device layer, the runtime system CoDeSys SP, has to be implemented." (WCP 10004)

71. WAGO's website describes its "WAGO SPEEDWAY 767" products as follows: "WAGO SPEEDWAY 767 is a modular, IP67-rated I/O system which is ideal for applications in

harsh environment conditions." (http://www.wago.us/products/276.htm). WAGO's website describes this product as follows: "SPEEDWAY is a cabinet-free automation platform that can be mounted directly on equipment to speed installation and reduce costs. This IP67 rated automation system eliminates the need for control enclosures and reduces wiring time with standard IP67 cables and connectors." (http://www.wago.us/products/680.htm). WAGO's website describes some of the "Extended Features" for this product as follows: "Fieldbus Independence – SPEEDWAY supports major fieldbus protocols including MODBUS, EtherNet I/P, Profibus, DeviceNet and CANopen[,] ... IEC61131-3 programmable controller in a IP67 form factor ... [and] 64 modules per node with a maximum 1,650ft node extension." (http://www.wago.us/products/680.htm). WAGO's website provides additional information about "Programmable Controllers and I/O Couplers" ("767-2301 Ethernet Programmable Controller" and "767-2501 CANopen Programmable Controller") and "I/O Module Overview" ("767-1101 Profibus DP/V1 Fieldbus Coupler," "767-1201 Profinet Fieldbus Coupler," "767-1301 Ethernet Fieldbus Coupler," "767-1401 DeviceNet Fieldbus Coupler," and "767-1501 CANopen Fieldbus Coupler"). (http://www.wago.us/products/39439.htm).

72. WAGO states that the 767 Speedway series of products contains a programmable bus coupler which is programmable with CoDeSys version 3.0. (See Decramer 30(b)(6) Dep. 79:7-80:2).

73. WAGO's website describes its "PERSPECTO – Monitors and Panels" products as "WAGO's comprehensive monitor and panel system for operating and monitoring process data for machines, systems and control technology." (http://www.wago.us/products/31270.htm). WAGO's website describes the "PERSPECTO CP" product as follows: "In addition to the HMI runtime, the control panel also has a CoDeSys runtime and is thus a full-fledged automation

device. It offers you configurable functions for operation and monitoring, and it can handle control tasks independently. The PLC functionality is based on the IEC 61131-compatible CoDeSys environment. With corresponding libraries, access to the hardware even from a PLC program is possible." (http://www.wago.us/products/31314.htm). WAGO's website states that the PERSPECTO CP product is available in the following formats: CP 35 QVGA – 3.5'' 320 x 240 pixels; CP 57 QVGA – 5.7'' 320 x 240 pixels; CP 104 VGA – 10.4'' 640 x 480 pixels; CP 121 SVGA – 12.1'' 800 x 600 pixels; and CP 150 XGA – 15'' 1024 x 768 pixels. (http://www.wago.us/products/31314.htm). WAGO's website describes the "PERSPECTO CP with Target VISU" product as follows: "*PERSPECTO*® CP with Target Visu for complete CoDeSys projects ... The Control Panel with Target Visu (TV) features full CoDeSys runtime. The CoDeSys development environment allows full application programming in CoDeSys. Existing projects may be almost entirely converted and upgraded. ... Standard CoDeSys libraries provide great compatibility and advanced solutions." (http://www.wago.us/products/37636.htm). WAGO's website states that the PERSPECTO CP with Target VISU product is available in the following formats: CP 35 QVGA – 3.5'' 320 x 240 pixels; CP 57 QVGA – 5.7'' 320 x 240 pixels; CP 104 VGA – 10.4'' 640 x 480 pixels; CP 121 SVGA – 12.1'' 800 x 600 pixels; and CP 150 XGA – 15'' 1024 x 768 pixels. (http://www.wago.us/products/37636.htm).

74. WAGO's website describes its "Condition Monitoring" products as follows: "Look-ahead maintenance enhances productivity and saves costs." (http://www.wago.us/products/276.htm). WAGO's website states that it offers "special measuring modules for the WAGO I/O System" through which "[s]ensor data is acquired in the modules and made available to higher-order evaluation units via standard fieldbuses. Data

evaluation is then conducted by means of standard visualization and process control systems available on the market." (http://www.wago.us/products/17780.htm).

75. WAGO's website describes its "Interface Modules" products as follows: "Besides pluggable function and interface modules, slim relays and optocouplers, as well as modern switch mode power supplies, WAGO also offers interface modules in different versions." (http://www.wago.us/products/276.htm). WAGO's website describes several different types of products that it offers: "Series 857," "Series 858," "Interface Modules for System Wiring," "Series 859," "Series 286," "Series 786," "Series 787," "Series 288, 287 and 289," "Series 789," "Series 788," and "Series 289." (http://www.wago.us/products/347.htm).

76. WAGO's website describes its "Overvoltage Protection" products as follows: "**Enhance reliability.** Used to protect other electrical equipment and systems against excess overvoltage and to generate equipotential bonding." (http://www.wago.us/products/276.htm). WAGO's website describes these products as "Surge protection devices" and explains that "[s]urge protection devices are electrical equipment whose principal components are voltage dependent resistors (varistors, suppressor diodes) and /or dischargers. Surge protection devices are used to protect other electrical equipment and systems against excess overvoltages and to generate equipotential bonding." (http://www.wago.us/products/345.htm).

77. WAGO's website describes its "Power Supplies" products as follows: "With primary switch mode power supply units, constant voltage sources, power supply and back-up capacitor modules, WAGO offers a comprehensive range of power supply units for a wide range of applications." (http://www.wago.us/products/276.htm). WAGO's website provides additional information about these products under the headings "EPSITRON® Power Supplies," "Power Supplies and Constant Voltage Sources for DIN 35 rail," "Power Supply Units," "Back-Up

Capacitor Module," "288 & 289 Series, DC / DC Converter," and "859 Series, DC / DC Converter." (http://www.wago.us/products/343.htm).

78. WAGO's website describes its "JUMPFLEX Signal Conditioners and Relays" products as follows: "The perfect match of housing and electronics is the key factor for a highly successful device. This is exactly what WAGO has achieved with the new 857 Series Signal Conditioners and Relay/Optocoupler Modules." (http://www.wago.us/products/276.htm).

79. Industrial control systems generally require multiple input and output modules and accessories to facilitate process connections.

80. A customer who uses a WAGO-I/O System industrial controller must also use WAGO-I/O System series 750 or 753 modules and accessories to accommodate directly interfaced local I/O.

81. In my experience, customers who use industrial automation controllers like the ones sold by WAGO generally use them to in a locally connected fashion. A typical configuration would include a controller, 2 to 10 I/O modules, an end module, and a power supply.

V ANALYSIS AND TESTING

82. I have been informed that the following WAGO-I/O System devices have been accused of infringing the asserted claims ("Accused Instrumentalities"):

a. WAGO-I/O-System 750;

b. WAGO-I/O-System 753;

c. WAGO-I/O-System 755;

d. WAGO-I/O-System 757;

e. WAGO-I/O-System 758; and

f. WAGO-I/O-PRO CAA.

83. I was provided with representative samples of the Accused Instrumentalities to be analyzed and tested in regards to infringement of the patents in dispute. The products were supplied by Plaintiffs' counsel.

84. The list of products that were provided included:

a. 750-410 (2 channel DI)

b. 750-501 (2 channel DO)

c. 750-841 (Modbus TCP controller)

d. 750-600 (end module)

e. 750-842 (Ethernet controller)

f. 750-872 (Ethernet controller, telecontrol)

g. 750-880 (Modbus TCP controller)

h. 758-870 (IPC)

i. 758-870/0000-0110 (I/O-IPC-G2)

j. 750-600 (8 DI-DO)

k. 759-911 (Wago I/O Pro CAA)

l. WAGO Automation Tool and Docs DVD (Rel. 019)

m. WAGO Automation Cockpit

n. HP notebook computer with Windows 7

85. Additional equipment provided by the author included:

a. Dell notebook used for initial testing

b. Ethernet network switch and cables

c. Misc. electronic test equipment and power supplies

d. Misc. LCD displays, computer keyboards and mice, USB storage devices.

86. The general test arrangement involved connection of WAGO IPC and 750 Controllers, each with WAGO I/O devices, to a personal computer (PC) using a local area network (LAN). The PC contained the WAGO-I/O-PRO CAA to program and operate the products on the LAN. This arrangement allowed analysis of the products operating in the operating environments described in WAGO documentation.

87. Not all of the supplied equipment was tested due to inadequate technical information, missing documentation, discrepancies with supplied documentation, or apparent similarity to other tested products.

88. Two control programs were developed and tested using WAGO-I/O-PRO CAA to demonstrate various features of WAGO products that are relevant to the patents in dispute. Details of the testing procedure are included as Exhibit C.

VI. MY OPINIONS

89. As noted above, references to "WAGO" include both WAGO Corporation and WAGO Kontakttechnik GmbH & Co. KG ("WAGO Germany"). Any instances requiring differentiation between the two Defendants will be appropriately noted.

90. I am not an attorney. Thus, for the purposes of this report, I have been informed about certain aspects of patent law that are relevant to my analyses and opinions.

91. The manuals, marketing materials, and materials publicly available on the WAGO websites are provided to WAGO customers in order to instruct them how to assemble, install, and operate the WAGO-I/O System Devices. WAGO anticipates and intends that its customers

will use these materials as WAGO instructs. In this way, WAGO has instructed its customers to infringe the Accused Instrumentalities.

A. The "Person of Ordinary Skill in the Art"

92. I am informed that there are some patent issues that require analysis from the perspective of a "person having ordinary skill in the art," or "PHOSITA."

93. It is my opinion that a PHOSITA during the time of the filing of the patent applications (2000-2002) would likely have:

a. a bachelor's degree (or equivalent practical experience) in engineering, computer science, or a related field;

b. would be capable of designing and operating hardware and software, and also related communications and support equipment, used for networked programmable industrial control applications;

c. and had at least five years of experience in working with automated process controls, industrial controls, and integrated systems.

94. My opinion regarding a PHOSITA is based on my 30+ years of experience developing hardware and software for equipment and systems closely related to the patents in dispute. (A detailed CV is included as Exhibit A to this report). My opinion is also based on study of the patents at issue, their prosecution histories, documents produced, the information publically available on WAGO's websites, testimony given during depositions in this lawsuit, my examination and testing of the WAGO products, and other information identified in this report.

B. Claim Construction

95. It is my understanding that the parties have exchanged lists of claim terms that each party contends requires interpretation, and the parties have exchanged their respective positions about, and support for, how the various claim terms should be interpreted. I have reviewed the following related documents:

a. "Plaintiff's Preliminary Proposed Claim Constructions" and "Exhibit A to Plaintiff's Preliminary Claim Constructions;"

b. "Defendants' List of Claim Terms and Proposed Constructions;"

c. "Plaintiffs' Amended Preliminary Proposed Claim Constructions" and "Exhibit A to Plaintiff's Amended Preliminary Claim Constructions;" and

d. Defendants' Responses and Objections to Plaintiffs' Third Set of Common Interrogatories along with Exhibit A.

96. It is my understanding that the Court will not be holding a separate hearing to determine the meaning of any of the claim terms that the parties have identified. Thus, in this report, I will present my opinions with regard to both sides' proposed claim constructions.

97. I have been informed by counsel for Rockwell that claim construction is a matter of law.

98. I have been informed by counsel for Rockwell that in construing claims, the intrinsic evidence of record, including the claims of the patent, the written description, and the prosecution history should be consulted. I should begin with the language of the claims and, in general, there is a heavy presumption in favor of the ordinary meaning of claim language as understood by one of ordinary skill in the art. It is also my understanding that this presumption

may be overcome where examination of the specification, prosecution history, and other claims indicates that the inventor intended otherwise.

99. Within this report, unless stated otherwise, I have applied each of the constructions advocated by Plaintiffs or by a person of ordinary skill in the art would at the time of the invention.

100. To the extent the Court adopts a specific claim that differs from one that I used for my opinion herein, I reserve the right to supplement the opinions contained herein.

101. My opinions in this report are based on my experience, my testing, the information disclosed in the documents listed in Exhibit B, the information on WAGO's websites, and based on other information in this report.

C. Defendants' Infringement

102. It is my understanding that patent infringement is the unauthorized making, using, selling, or offering to sell in the United States, or importing into the United States, a product or by using a method meeting all of the requirements of a claim in a United States patent owned by someone else.

103. It is my understanding that patent infringement is analyzed on a claim-by-claim basis, and that it is done in two steps. First, the claims of the patents are interpreted as they would be by a PHOSITA. Second, each claim is applied to the accused product(s) or method to determine if every element of the claim is present in the accused product(s) or method.

104. It is my understanding that there are two kinds of infringement: direct infringement and indirect infringement.

105. It is my understanding that there are two types of direct infringement: literal infringement and infringement under the doctrine of equivalents.

106. It is my understanding that literal infringement occurs when each and every limitation of a patent claim is present in the accused product or method. It is my understanding that the inclusion of additional elements in an accused product or additional steps in an accused method does not negate literal infringement.

107. It is my understanding that if every element of a claim is not present in the accused product or process, the claim might still be infringed under the "doctrine of equivalents."

108. It is my understanding that under the doctrine of equivalents, a product or method infringes a claim if the accused product or method contains elements or performs steps corresponding to each and every requirement of the claim that is equivalent to, even though not literally present, the accused product or method.

109. It is my understanding that for the purposes of the doctrine of equivalents, a structure or action (i.e. the "aspect") is equivalent if a PHOSITA in the field of the patent either would have considered the differences between the accused aspect and the related aspect described in the patent as being insubstantial, or would have found that the aspect in the accused product or method (a) performs substantially the same function and (2) works in substantially the same way (3) to achieve substantially the same result as the requirement of the claim. It is my understanding that for a structure or action to be deemed interchangeable, it must have been known to a person of ordinary skill in the art at the time of infringement.

110. It is my understanding that claims can be written in "means-plus-function" format. It is my understanding that a product or method meets a means-plus-function claim element if (a) it has a structure or set of structures or an action or set of actions that performs the identical function recited in the claim, and (b) the structure or set of structures or the action or set

of actions is either identical or equivalent to the structures or actions disclosed in the patent that perform that function.

111. It is my understanding that, for the analysis of whether something is an equivalent for the purpose of determining infringement of a means-plus-function claim, a structure or action (i.e. "aspect") is equivalent if a PHOSITA in the field of the patent would have considered the differences between the accused aspect and the related aspect described in the patent as being insubstantial. I understand that in determining whether a difference is "insubstantial," one may consider whether a PHOSITA would have known of the interchangeability of the two aspects. I also understand that for an aspect to be deemed interchangeable, it must have existed at the time the patent issued.

112. It is my understanding that there are two kinds of indirect infringement: inducement and contributory.

113. I have been informed by counsel for Plaintiffs that indirect infringement requires a threshold showing of direct infringement. Direct infringement can be established by showing specific instances of direct infringement or by showing that the accused product or process necessarily infringes the patent claim. I understand that either type of direct infringement—whether specific instances or necessarily infringing elements—can be proven with direct or circumstantial evidence. Circumstantial evidence may include sales data for the accused product, results of user trials, instructional materials that teach users to infringe, and advertising or marketing materials concerning infringing features.

114. It is my understanding that a company or a person can be found to have actively induced infringement if (a) the company or person took actions specifically intending to cause acts by a direct infringer that resulted in direct infringement of the patent claim, (b) the company

or person was aware of the patent and knew or should have known that the acts, if taken by the direct infringer, would constitute infringement, and (c) the acts were or are carried out by the direct infringer and directly infringed or infringe the claim.

115. I am told that for the purposes of inducement, intent may be inferred from circumstantial evidence, for example, the accused infringer's knowledge of the patent and control over manufacture.

116. It is my understanding that a company or a person can be found to have committed contributory infringement if (a) the company or person sells, offers to sell, or imports within the United States a component of a product, or apparatus for use in a method, (b) the component or apparatus has no substantial, non-infringing use, (c) the component or apparatus constitutes a material part of the invention, (d) the company or person is aware of the patent and knows that the products or methods for which the component or apparatus has no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and (e) that use directly infringes the claim.

117. Additionally, I understand that an accused infringer may not escape liability for contributory infringement by merely embedding a component that lacks a non-infringing use into a larger product.

118. In the following sections, I present my opinions with regard to each of the asserted claims of each of the patents at issue. For ease of reference, I have inserted numbers in brackets at the start of each claim element to help identify each of the elements in claims having more than one claim element.

119. The Accused Instrumentalities include the WAGO-I/O System devices that are made, used, offered for sale, sold or imported in to the United States.

120. The Accused Instrumentalities include:

a. Infringing Software = WAGO-I/O-PRO CAA

b. Infringing PLCs = All of the WAGO-I/O System PLCs with attached I/O modules that are programmable with the Infringing Software. (See CoDeSys Compatibility Chart)

c. Infringing File System PLCs = Infringing PLCs that have file system services (the WAGO PLC that was first sold that meets this definition is the 750-841, which I am told was released sometime in 2003). (See Albers 30(b)(6) Dep. Tr. 73:22-25).

121. It is my understanding that there are very few meaningful differences between the Infringing PLCs, at least insofar as infringement is considered for certain asserted patents in this case. (See, e.g., Albers and Decramer Depositions and January 2012 Letter from WAGO's counsel).

122. It is my understanding that there are very few meaningful differences between the Infringing File System PLCs, at least insofar as infringement is considered for certain asserted patents in this case. (Id.)

123. It is my understanding that there are very few meaningful differences between versions of the Infringing Software, at least insofar as infringement is considered for certain asserted patents in this case. (Id.)

124. I have reviewed WAGO's noninfringement position as stated in its responses to discovery requests. I consider certain of WAGO's noninfringement positions to be vague and/or incorrect, including at least with respect to whether the Accused Instrumentalities read on the

limitations of the claims. As such, I reserve the right to modify or supplement my opinion based upon future disclosures, including any noninfringement report prepared on behalf of WAGO.

125. In the following sections, I present my infringement analysis with regards to the elements of each of the asserted claims. While the headings in this report are used for clarity, they are not meant to limit or otherwise restrict my testimony. I reserve the right to testify with regards to infringement of the asserted claims based on any combination of the statements disclosed in this report.

126. In light of the above, unless I indicate otherwise below, my statements and opinions apply to all Accused Instrumentalities equally in each of the configurations identified above.

1. WAGO's Infringement of U.S. Patent No. 6,745,232

127. I understand that Plaintiffs are asserting claims 1, 2, 3, 5, 10, 11, and 14 of the '232 patent. It is my opinion that WAGO has been and is infringing claims 1, 2, 3, 5, 10, 11, and 14 of the '232 patent, has induced and is inducing infringement of those claims, and has contributed and is contributing to their customers' infringement of those claims.

128. Based on my review, I believe that the Infringing PLCs practice each and every element of claims 1, 2, 3, 5, 10, 11, and 14 of the '232 patent. My opinion is based on the materials I reviewed, my tests, and my experience.

129. I do not believe any of the claim terms in the '232 patent require explanation or a Court's interpretation, as the terms are clear and easily understood not only by a PHOSITA, but simply by laypeople as well. It is my opinion that the terms of the '232 patent are understood in light of the patent's specification and the claims themselves, such that no construction should be adopted that alters the plain meaning of these words.

130. While the term "control device" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt a meaning for control device, it should adopt Plaintiffs' definition which supports a definition for "control device" to mean a PLC. This construction of the term "control device" is supported by the intrinsic evidence in the '232 patent and its prosecution history. However, it is my opinion that the claim elements are satisfied for infringement using either party's construction of the term "control device."

131. While the term "communications medium" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience, and because Plaintiffs' position is supported by the intrinsic evidence in '232 patent and its prosecution history. However, it is my opinion that the claim elements are satisfied for infringement using either party's construction of the term "communications medium."

132. While the term "mode change message" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience, and because Plaintiffs' position is supported by the intrinsic evidence in the '232 patent and its prosecution history. I note that Defendants position is not supported by the intrinsic evidence of the '232 patent and attempts to read limitations in the claim term "mode change message" that should not be included. Moreover, I have not been provided with the development source code or other information that would allow me to understand how the mode change

message in the WAGO products is specifically implemented, therefore I am unable to give an opinion on whether the WAGO products infringe the claims of the '232 patent based on Defendants' definition of mode change message.

133. While the term "step mode" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience, and because Plaintiffs' position is supported by the intrinsic evidence in the '232 patent and its prosecution history. However, taking either party's construction of the term "step mode," it is my opinion that the claim elements are satisfied for infringement.

134. While the term "mode change message with instructions therein to execute the stored program in a step mode from the location in which the program was suspended" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience, and because Plaintiffs' position is supported by the intrinsic evidence in '232 patent and its prosecution history. I note that Defendants position is not supported by the intrinsic evidence of the '232 patent and attempts to read limitations in the claim term "mode change message" that should not be included. Moreover, I have not been provided with the development source code or other information that would allow me to understand how the mode change message with instructions therein to execute the store program in a step mode from the location in which the program was suspended in the WAGO products is specifically implemented, therefore I am unable to give an opinion on whether the WAGO

products infringe the claims of the '232 patent based on Defendants' definition of "mode change message with instructions therein to execute the store program in a step mode from the location in which the program was suspended."

135. While the term "processor" as used in the '232 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction because Plaintiffs' position is supported by the intrinsic evidence in '232 patent and its prosecution history. However, taking either party's construction of the term "processor," it is my opinion that the claim elements are satisfied for infringement.

a. Claim 1

136. Claim 1 reads as follows:

> 1. A method of performing a function in a control device comprising:
>
> [1] storing a program in a control device, the control device receives a message from a communications medium, the message includes instructions to suspend execution of the stored program at a particular location of the stored program;
>
> [2] executing at least a portion of the stored program in the control device according to the instructions;
>
> [3] suspending execution of the stored program according to the instructions; and
>
> [4] receiving a mode change message with instructions therein to execute the stored program in a step mode from the location in which the program was suspended.

137. Claim 1 of the '232 patent is directed to "A method of performing a function in a control device comprising:". I have been informed that this portion of the claim is called the preamble and does not give life and meaning to the claims, and therefore does not serve to limit

the claims. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion that WAGO does practice a method for performing a function in a control device.

(i) Element [1] of Claim 1

138. Claim 1 includes the element, "storing a program in a control device, the control device receives a message from a communications medium, the message includes instructions to suspend execution of the stored program at a particular location of the stored program." The Infringing PLCs store a program and receive a message from a communications medium that includes instructions to suspend execution of the stored program at a particular location of the stored program.

139. Specifically, element [1] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents, by the Infringing PLCs for at least the following reasons:

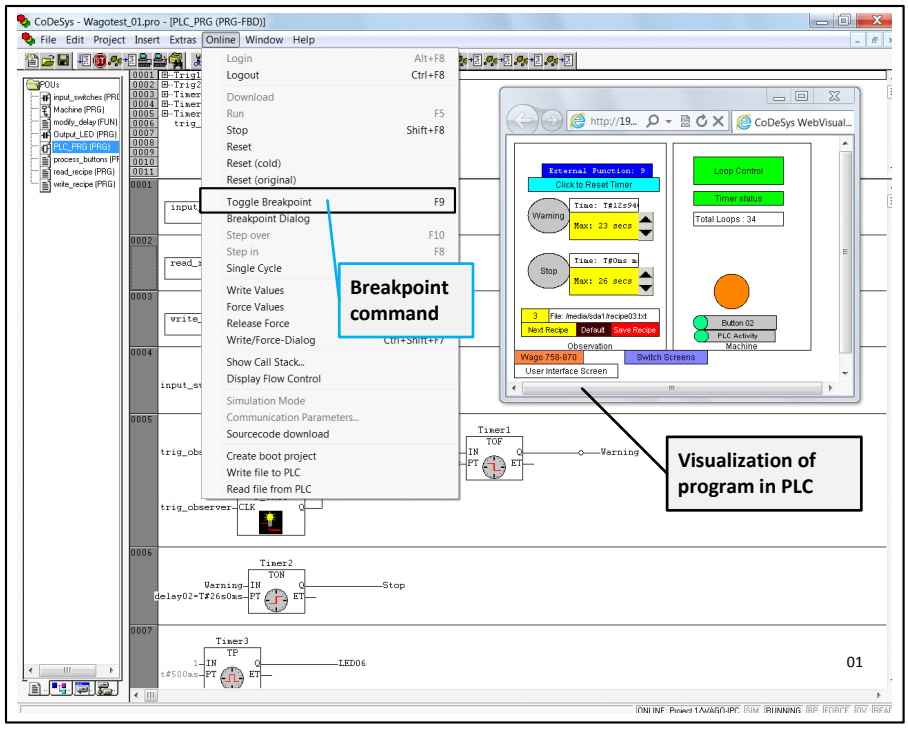
a. An Infringing PLC is a control device that is capable of being programmed with the Infringing Software. (See WAGO-I/O-PRO CAA Compatibility Chart [ROCK0007738-ROCK0007744]; CoDeSys 2.3 Manual [ROCK0007278-ROCK0007737]; 750-841 Manual Section 3.1.7 [ROCK0008463]; 758-870/000-110 Manual Section 11-12 [ROCK0008687-ROCK0008720]).

b. A program created using the Infringing Software can be stored in the Infringing PLC. (See CoDeSys 2.3 Manual [ROCK0007410]; Albers 30(b)(6) Dep. Tr. 66:1-4, 86:2-4).

c. The Infringing PLC receives a message from a PC running the Infringing Software via a communications medium (e.g. using Ethernet link in testing).

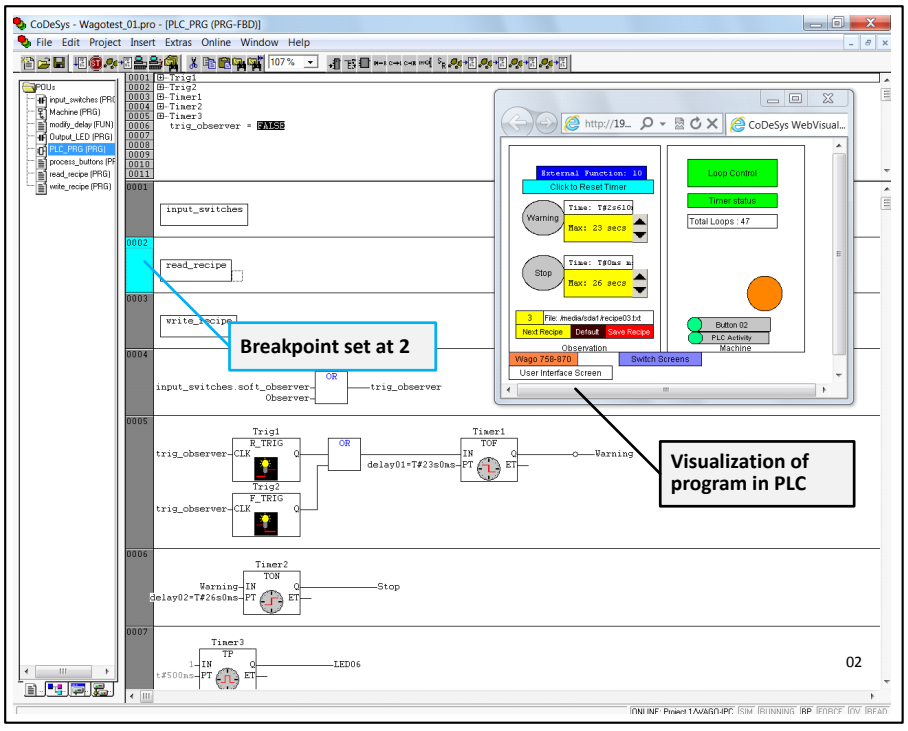
d. The message received by the Infringing PLC contains instructions to suspend execution of the program running on the PLC at a particular program location. This message was generated during my testing by clicking a specific program instruction within the access tool running on the PC to indicate where the execution will be suspended in the PLC. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs store a program and receive a message from a communications medium that includes instructions to suspend execution of the stored program at a particular location of the stored program as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

e. The Infringing Software has an online mode that allows sending a breakpoint command to suspend execution of a stored program on a Infringing PLC at a particular location of the stored program:



(Testing 758-870 IPC Controller)

f. The screenshot below shows that a break point (depicted in light blue) has been set at instruction line 2:



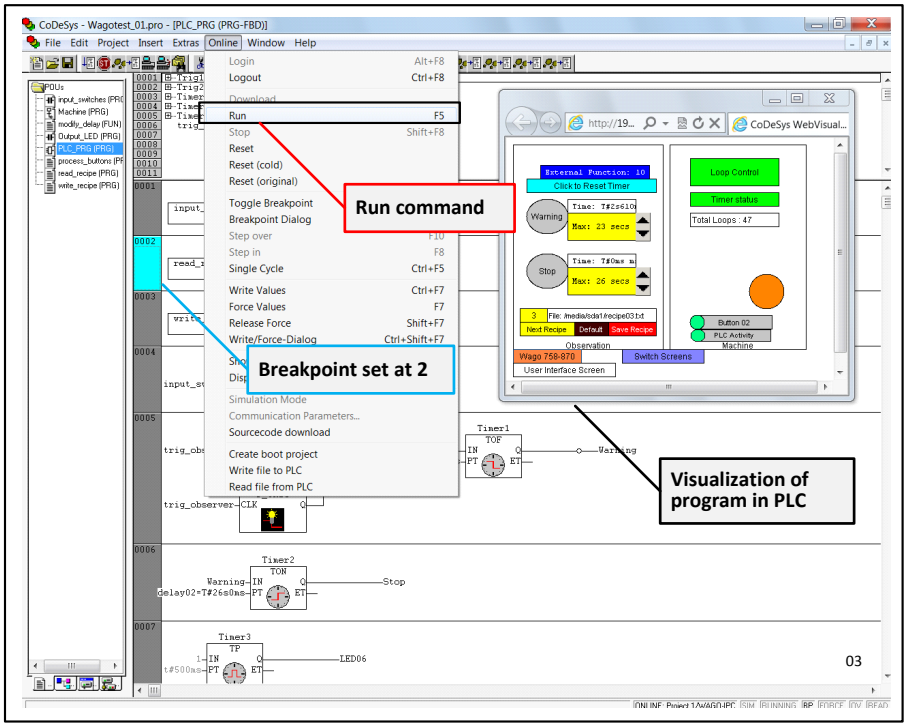
(Testing of CoDeSys 2.3 with WAGO 758-870 Controller)

(ii) Element [2] of Claim 1

140. Element [2] of Claim 1 includes the element, "executing at least a portion of the stored program in the control device according to the instructions." The Infringing PLCs execute at least a portion of the stored program according to the instructions where no breakpoint has been set in the PLC program.

141. Specifically, element [2] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents, by the Infringing PLCs for at least the following reasons:

a. In the screenshot below, the Infringing PLC receives a message from a communications medium (selection of the "Run" command) to start execution of a stored program:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

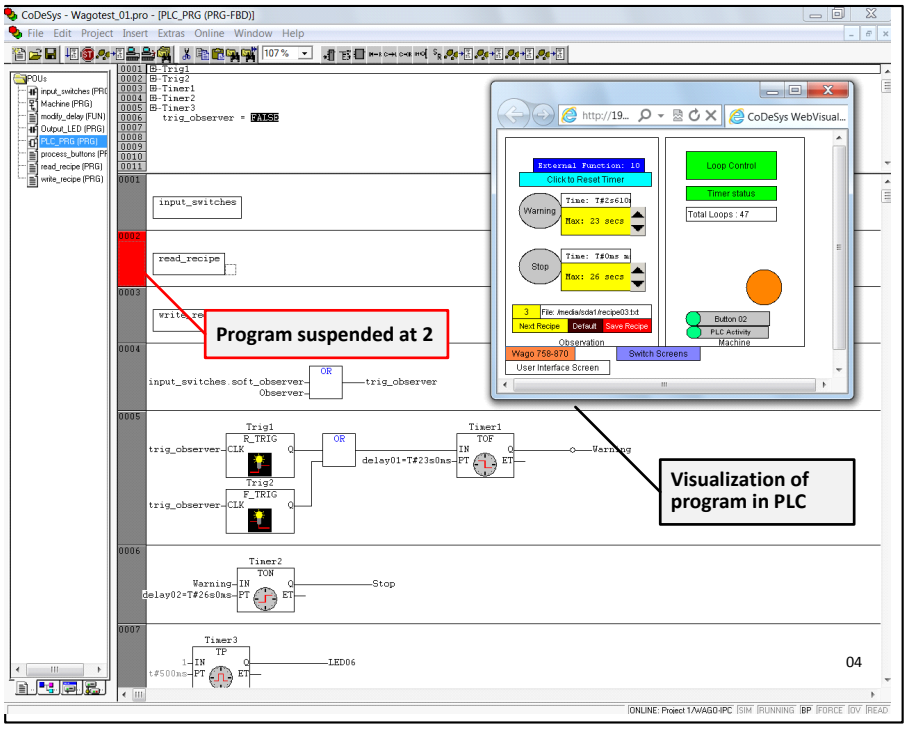
In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs execute at least a portion of the stored program according to the instructions when no breakpoint has been set in the PLC program as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(iii) Element [3] of Claim 1

142. "suspending program according to the instructions". The Infringing PLCs suspends execution of the stored program according to the instructions.

143. Specifically, element [3] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents, by the Infringing PLCs for at least the following reasons:

a. The screenshot below shows suspending execution of a stored program running on an Infringing PLC with the red block showing where the execution of the stored program has been suspended at instruction line 2 after executing instruction line 1:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs suspend execution of the stored program according to the instructions as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

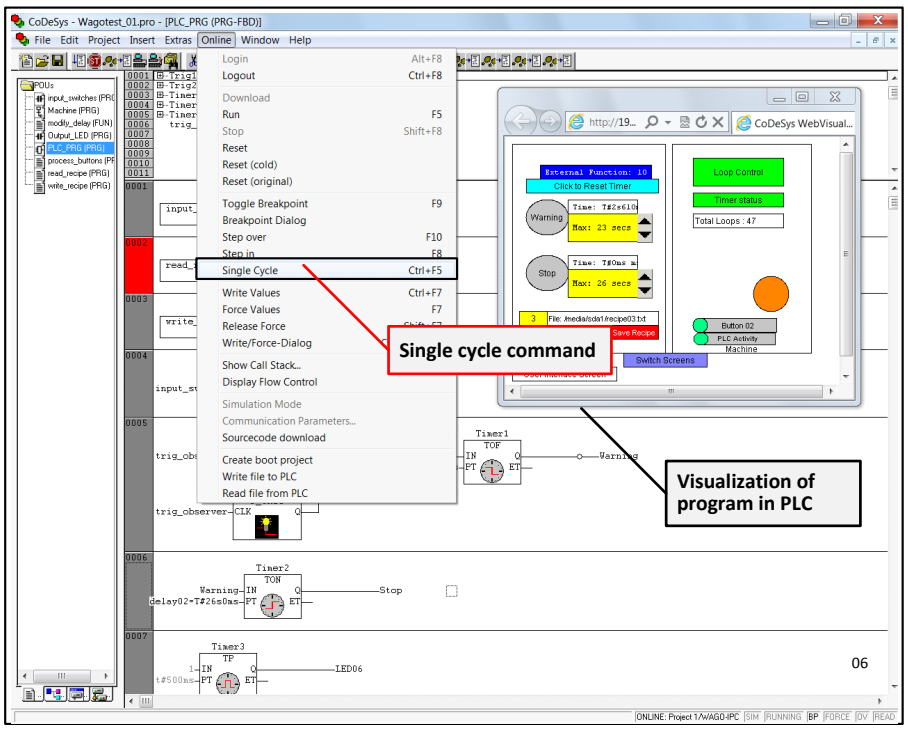
(iv) Element [4] of Claim 1

144. Element [4] of Claim 1 includes the element, "receiving a mode change message with instructions therein to execute the stored program in a step mode from the location in which the program was suspended." The Infringing PLCs receive a mode change message with

instructions therein to execute the stored program in a step mode from the location in which the program was suspended.

145. Specifically, element [4] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents, by the Infringing PLCs for at least the following reasons:

a. The Infringing PLC receives a mode change message, e.g. the "Online" drop-down menu has several mode change options (e.g., "Single Cycle"). These mode change options generate a message containing instructions received by the Infringing PLC to execute a least a portion of the stored program according to the step mode from the location where the program was suspended. The screen shot below shows the selection of a Single Cycle to operate the Infringing PLC in a step mode:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

> In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs receive a mode change message with instructions therein to execute the stored program in a step mode from the location in which the program was suspended as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

146. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 1 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

147. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

148. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of WAGO's invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

149. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 1 of the '232 patent.

150. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 1 of the '232 patent.

b. Claim 2

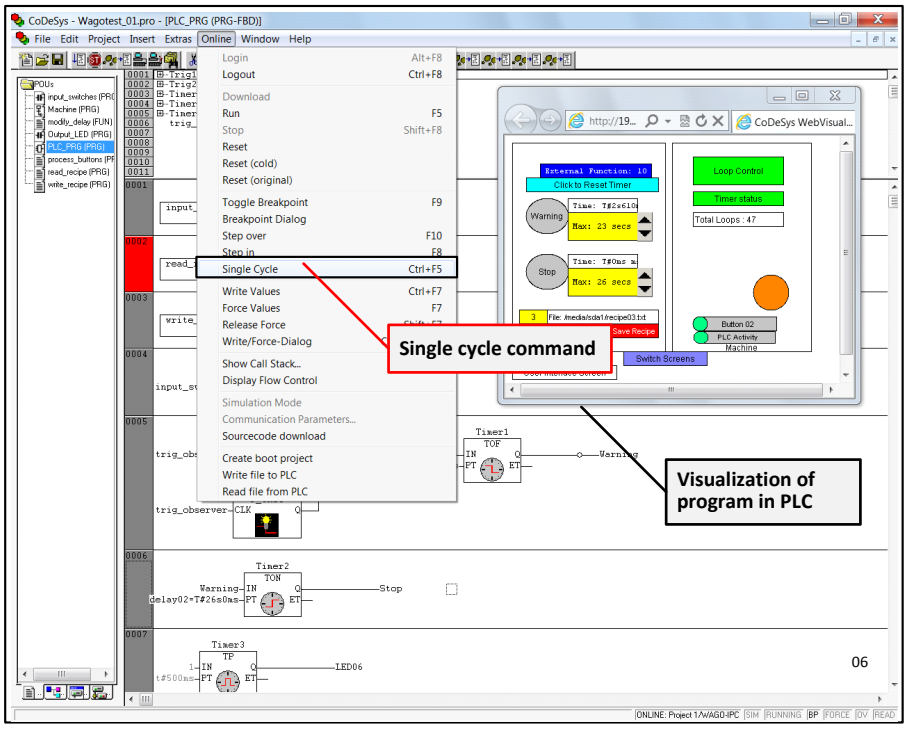
151. Claim 2 reads as follows:

> 2. The method of claim 1, further comprising repeating executing the at least a portion of the stored program and suspending execution of the stored program, in response to another message from the communications medium.

152. The Infringing PLCs repeat execution of a portion of the stored program and suspend execution of the stored program, in response to another message from the communications medium.

153. Specifically, Claim 2 is met by the Infringing PLCs literally, or in the alternative, under the doctrine of equivalents, for at least the following reasons:

a. The foregoing screenshots and the screenshot below show that the execution of a stored program in the Infringing PLC can be repeated and the suspension of execution of the stored program can be repeated in response to another message from the communications medium (i.e. "single cycle"). This was observed in testing.



(Testing on WAGO 758-870 IPC Controller)

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs repeat execution of a portion of the stored program and suspend execution of the stored program, in response to another message from the communications medium as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

154. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 2 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

155. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

156. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of Plaintiffs' invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

157. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of claim 2 of the '232 patent.

158. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 2 of the '232 patent.

c. Claim 3

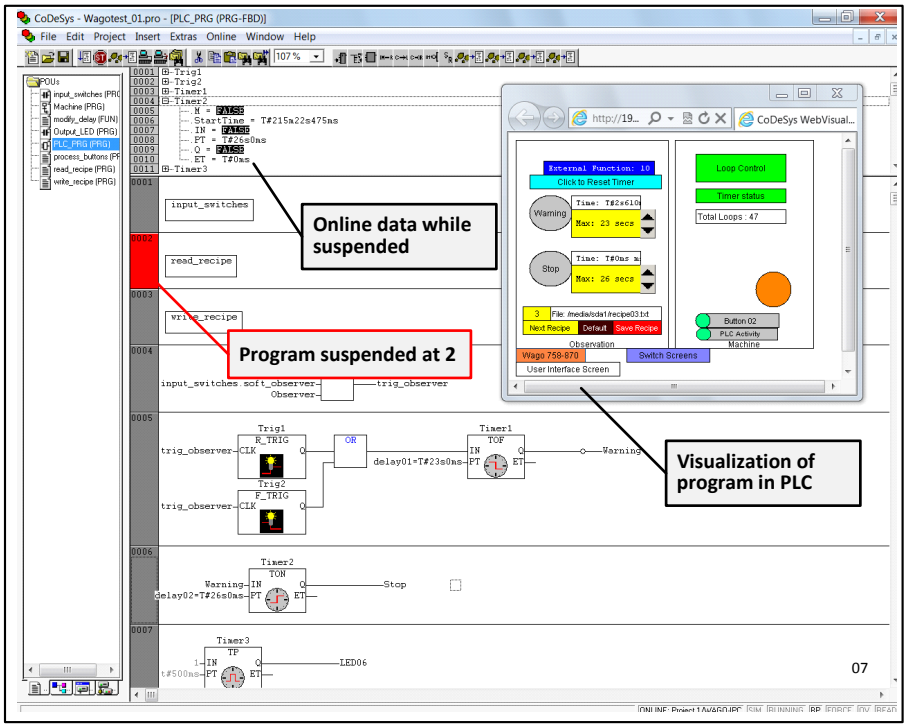
159. Claim 3 reads as follows:

> 3. The method of claim 1, further comprising providing data to the communications medium in response to a data request message from a network while execution of the stored program is suspended.

160. The Infringing PLCs provide data to the communications medium in response to a data request message from a network while execution of the stored program is suspended.

161. Specifically, Claim 3 is met literally, or in the alternative, under the doctrine of equivalents, by the Infringing PLCs for at least the following reasons:

a. The screenshot below shows a stored program on an Infringing PLC has been suspended at instruction line 2, and the Infringing PLC provides PLC memory data to the communications medium in response to a data request message from the online access tool (i.e. WAGO-I/O-PRO CAA) over an Ethernet link while execution of the stored program is suspended:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs provide data to the communications medium in response to a data request message from a network while

execution of the stored program is suspended as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

162. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 3 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

163. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

164. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of Plaintiffs' invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

165. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of claim 3 of the '232 patent.

166. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 3 of the '232 patent.

d. Claim 5

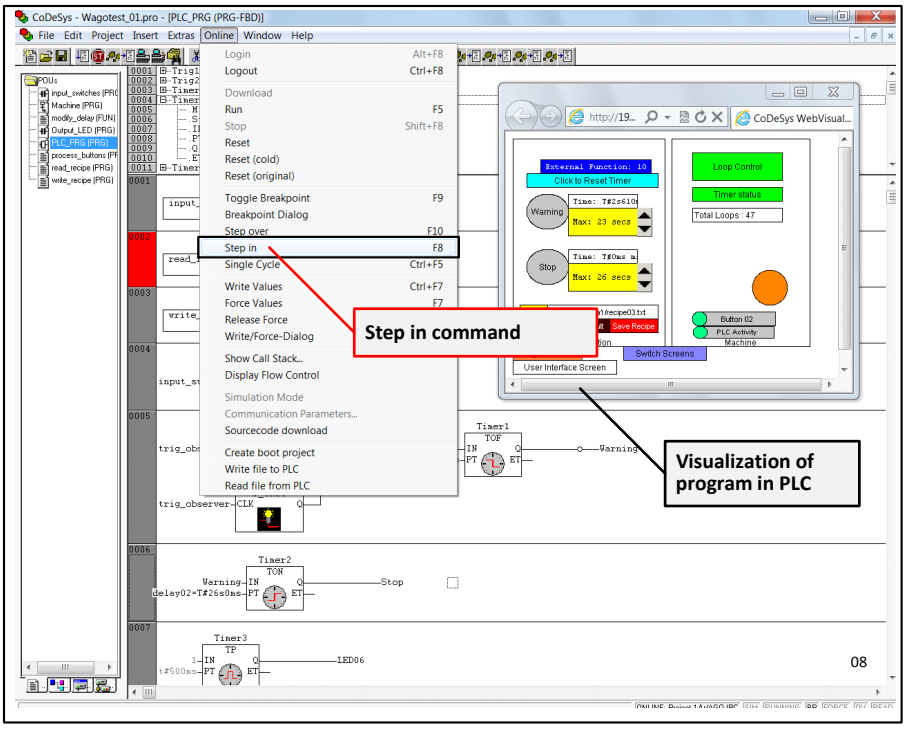
167. Claim 5 reads as follows:

> 5. The method of claim 1, wherein the mode change message further comprises a step type, and wherein executing the at least a portion of the stored program and suspending execution of the stored program are done according to the step type.

168. The mode change message that is received by the Infringing PLC comprises a step type, and wherein executing the at least a portion of the stored program and suspending execution of the stored program are done according to the step type.

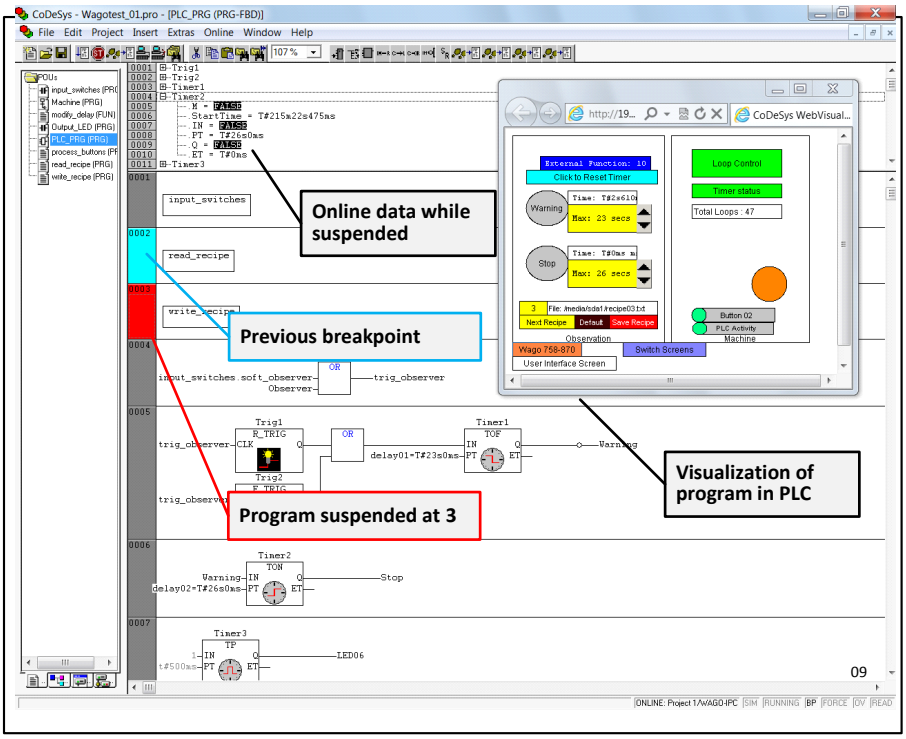
169. Specifically, Claim 5 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The Infringing PLC receives a mode change message, e.g. the "Online" drop-down menu has several mode change options (e.g., "Step In"). These mode change options generate a message containing instructions received by the Infringing PLC to execute a least a portion of the stored program according to the step mode from the location where the program was suspended. The screen shot below shows the selection of the step type "Step In" and my testing confirms that the Infringing PLC executed the program according to that step type:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

After executing a portion of the program based on the "Step In" command from the breakpoint at instruction line 2, the controller stops execution at instruction line 3. (Other step types besides "step in" could also be selected).



In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the mode change message generated by the Infringing Software that is received by the Infringing PLCs comprises a step type, and because executing at least a portion of the stored program and suspending execution of the stored program are done according to the step type as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

170. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 5 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

171. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

172. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of Rockwell's invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

173. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 5 of the '232 patent.

174. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 5 of the '232 patent.

e. Claim 10

175. Claim 10 reads as follows:

> 10. The method of claim 5, further comprising repeating executing the at least a portion of the stored program and suspending execution of the stored program, in response to another message from the communications medium.

176. The Infringing PLCs repeat a step type, and wherein executing the at least a portion of the stored program and suspending execution of the stored program are done according to the step type.

177. Specifically, Claim 10 is met by the Infringing PLCs literally, or in the alternative, under the doctrine of equivalents, for at least the following reasons:

a. The foregoing screenshots show that the execution of a stored program in the Infringing PLC can be repeated and the suspension of execution of the stored program can be repeated in response to another message from the communications medium (i.e. "single cycle"). This was observed in testing. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs repeat a step type and execute at least a portion of the stored program and suspend execution of the stored program according to the step type as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

178. Based on the foregoing, and based on other information disclosed in this report, it is my opinion WAGO has infringed and is infringing Claim 10 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

179. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are

using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

180. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of Rockwell's invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

181. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 10 of the '232 patent.

182. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 10 of the '232 patent.

f. Claim 11

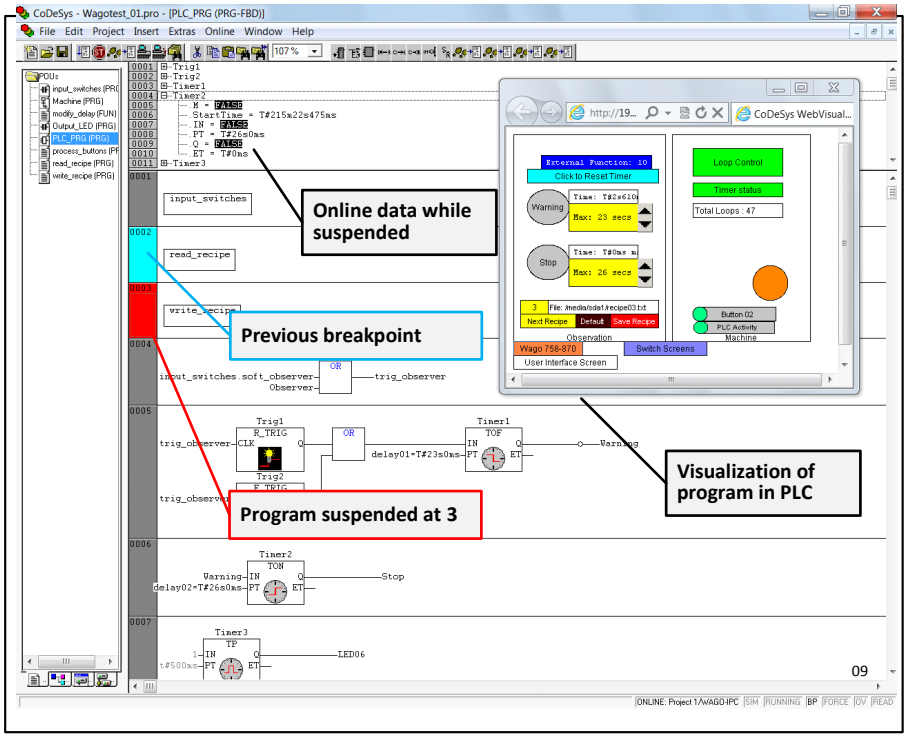
183. Claim 11 reads as follows:

> 11. The method of claim 5, further comprising providing data to the communications medium in response to a data request message from the network while execution of the stored program is suspended.

184. The Infringing PLCs provide data to the communications medium in response to a data request message from a network while execution of the stored program is suspended.

185. Specifically, Claim 11 is met by the Infringing PLCs literally, or in the alternative, under the doctrine of equivalents for at least the following reasons:

a. The screenshot below shows a stored program on an Infringing PLC has been suspended and the Infringing PLC provides PLC memory data to the communications medium in response to a data request message from the online access tool (i.e. WAGO-I/O-PRO CAA) over an Ethernet link while execution of the stored program is suspended at instruction line 3:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs provide data to the communications medium in response to a data request message from a network while execution of the stored program is suspended as set forth herein, and thus

perform substantially the same function in substantially the same way to yield substantially the same result.

186. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 11 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

187. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of claim 11 of the '232 patent.

188. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 11 of the '232 patent.

g. Claim 14

189. Claim 14 reads as follows:

> 14. A method of performing a function in a control device comprising:
>
> [1] providing a control device that selectively executes a program and receives messages from a network;
>
> [2] receiving a mode change message from the network;
>
> [3] suspending execution of the program according to the mode change message;
>
> [4] receiving a step command message from the network;
>
> [5] executing at least a portion of the program in the control device according to the message; and
>
> [6] suspending execution of the program according to the message.

190. Claim 14 of the '232 patent is directed to "A method of performing a function in a control device comprising". I have been informed that this portion of the claim is called the

preamble and does not give life and meaning to the claims, and therefore does not serve to limit the claims. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion that WAGO does practice a method for performing a function in a control device.

(i) Element [1] of Claim 14

191. Element [1] of Claim 14 includes, "providing a control device that selectively executes a program and receives messages from a network;" The Infringing PLCs selectively execute a program and receive messages from a network.

192. Specifically, element [1] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons.

a. An Infringing PLC is a control device that is capable of being programmed with the Infringing Software.

b. An Infringing PLC can selectively execute a program in the PLC.

c. A program created using the Infringing Software can be stored in the Infringing PLCs.

d. The Infringing PLC receives a message from a PC running the Infringing Software via a network (e.g. using Ethernet link in testing).

e. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs selectively execute a program and receive messages from a network as set forth herein, and thus perform

substantially the same function in substantially the same way to yield substantially the same result.

(ii) Element [2] of Claim 14

193. Element [2] of Claim 14 includes the element, "receiving a mode change message from the network;". The Infringing PLCs receive a mode change message from the network.

194. Specifically, element [2] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The message received by the Infringing PLC is a mode change message from the network. This was observed during testing by stopping the execution of a program on an Infringing PLC.

b. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs receive a mode change message from the network as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

(iii) Element [3] of Claim 14

195. Element [3] of Claim 14 includes the element, "suspending execution of the program according to the mode change message;". The Infringing PLCs suspend execution of the stored program according to the mode change message.

196. Specifically, element [3] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The program executing on the Infringing PLC is suspended according to the mode change message. This was observed during testing by stopping the execution of a program on an Infringing PLC.

b. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs suspend execution of the stored program according to the mode change message as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

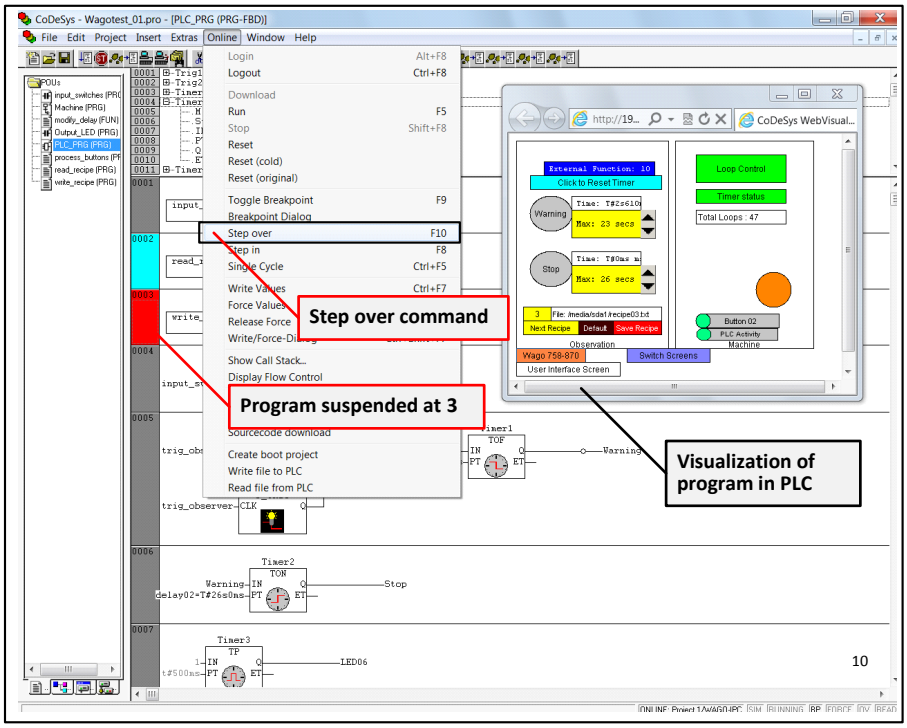
(iv) Element [4] of Claim 14

197. Element [4] of Claim 14 includes the element, "receiving a step command message from the network." The Infringing PLCs receives a step command message from the network.

198. Specifically, element [4] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The Infringing PLC receives a step command message, e.g. the "Online" drop-down menu has several step command messages (e.g., "Step Over"). These step command options generate a message containing instructions received by the Infringing PLC to execute a least a portion of the stored program according to the step command from the location where the program was suspended. The screen shot below shows the selection of a

step command, and my testing confirms that the instruction was received by the Infringing PLC to operate according to the step command:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller)

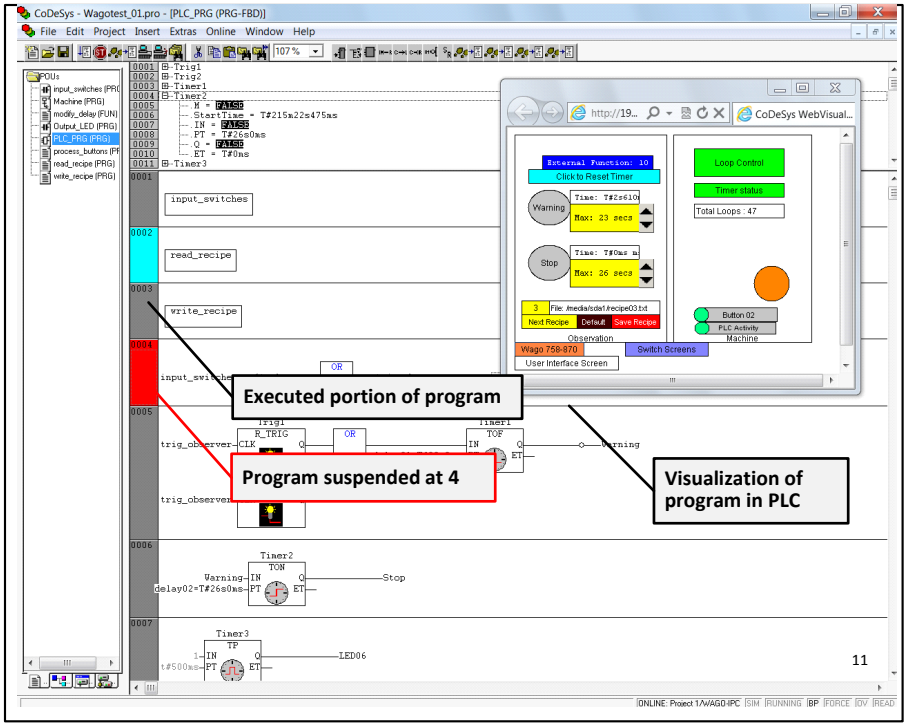
In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs receive a step command message from the network as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

(v) Element [5] of Claim 14

199. Element [5] of Claim 14 includes the element, "executing at least a portion of the program in the control device according to the message;". The Infringing PLCs execute at least a portion of the program in the Infringing PLC according to the message.

200. Specifically, element [5] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The Infringing PLC executes according to the step command. My testing confirms that the Infringing PLC executed the program according to the step command to "step over" depicted in the preceding screenshot.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

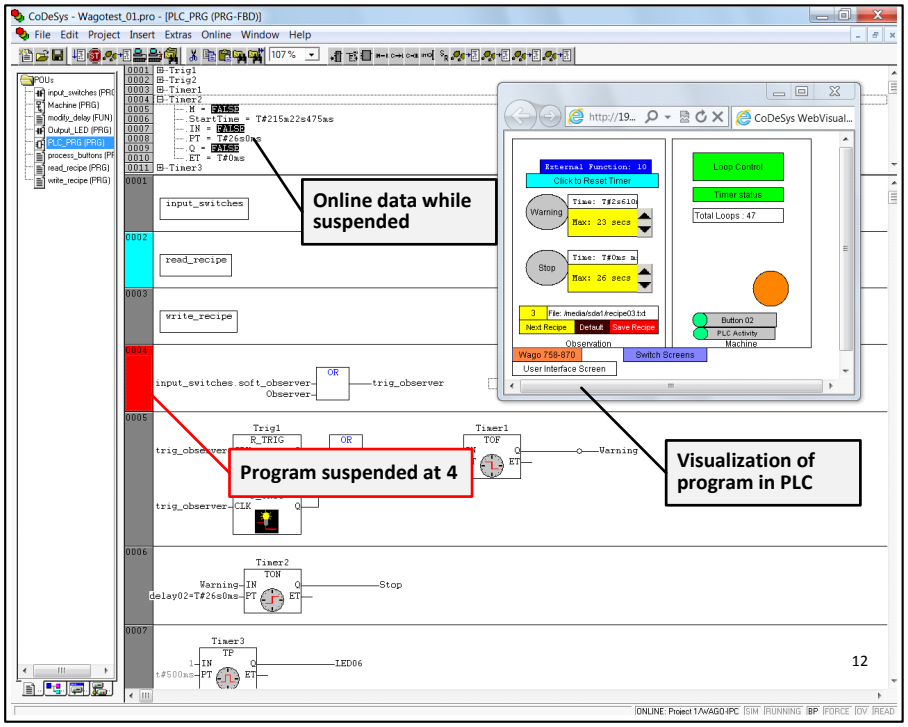
In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs execute at least a portion of the program in the Infringing PLC according to the message as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

(vi) Element [6] of Claim 14

201. Element [6] of Claim 14 includes the element, "suspending execution of the program according to the message." The Infringing PLCs suspend execution of the program according to the message.

202. Specifically, element [6] of Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing PLCs for at least the following reasons:

a. The screenshot below shows suspending execution of a stored program running on an Infringing PLC (red block) as a result of receiving a step command message (e.g. single cycle).



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

> In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing PLCs suspend execution of the program according to the message as set forth herein, and thus perform substantially the same function in substantially the same way to yield substantially the same result.

203. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 14 of the '232 patent literally, or in the alternative, under the doctrine of equivalents.

204. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

205. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, have no substantial, non-infringing uses and constitute a material part of Rockwell's invention. WAGO is aware of the '232 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '232 patent.

206. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 14 of the '232 patent.

207. Based on the foregoing, my testing, my experience, the information disclosed in the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 14 of the '232 patent.

2. WAGO's Infringement of U.S. Patent No. 7,123,974

208. I understand that Plaintiffs are asserting claims 1-3, 5, 6, 9, 10, 14, 16, 24, and 29 of the '974 patent. It is my opinion that WAGO has been and is directly infringing claims 1-3, 5, 6, 9, 10, 14, 16, 24, and 29 of the '974 patent, has induced and is inducing infringement of Claim 24 of the '974 patent, and has contributed and is contributing to their customers' infringement of Claim 24, either literally or under the doctrine of equivalents.

209. Based on my review, I believe that the Accused Instrumentalities practice each and every element of claims 1-3, 5, 6, 9, 10, 14, 16, 24, and 29 of the '974 patent. My opinion is based on the materials I reviewed, my tests, my experience and other information in this report.

210. Other than the means plus function elements, I do not believe any of the claim terms in the '974 patent require explanation or a Court's interpretation, as the terms are clear and easily understood not only by a PHOSITA, but simply by laypeople as well. It is my opinion that the terms of the '974 patent are understood in light of the patent's specification and the claims themselves, such that no construction should be adopted that alters the plain meaning of these words.

211. While the term "schema" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the

industry, and because Plaintiffs' position is supported by the intrinsic evidence of the '974 patent and its prosecution history. However, taking either party's construction of the term "schema," it is my opinion that the term "schema," it is my opinion that the claim elements are satisfied for infringement.

212. While the term "audit data" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence of the '974 patent and its prosecution history. However, taking either party's construction of the term "audit data," it is my opinion that the claim elements are satisfied for infringement.

213. While the term "control components" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "control components," it is my opinion that the claim elements are satisfied for infringement.

214. While the term "data field" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the

industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "data field," it is my opinion that the claim elements are satisfied for infringement.

215. While the term "industrial control components" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "industrial control components," it is my opinion that the claim elements are satisfied for infringement.

216. While the term "industrial control environment" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "industrial control environment," it is my opinion that the claim elements are satisfied for infringement.

217. While the term "industrial controller" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my

knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. Plaintiffs' position supports the notion that an industrial controller in the context of the patent means a PLC. However, taking either party's construction of the term "industrial controller," it is my opinion that the claim elements are satisfied for infringement.

218. While the term "industrial controller system" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "industrial controller system", it is my opinion that the claim elements are satisfied for infringement.

219. While the term "processor" as used in the '974 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my knowledge of the industry, and because Plaintiffs' position is supported by the intrinsic evidence in the '974 patent and its prosecution history. However, taking either party's construction of the term "processor," it is my opinion that the claim elements are satisfied for infringement.

a. <u>Claim 1</u>

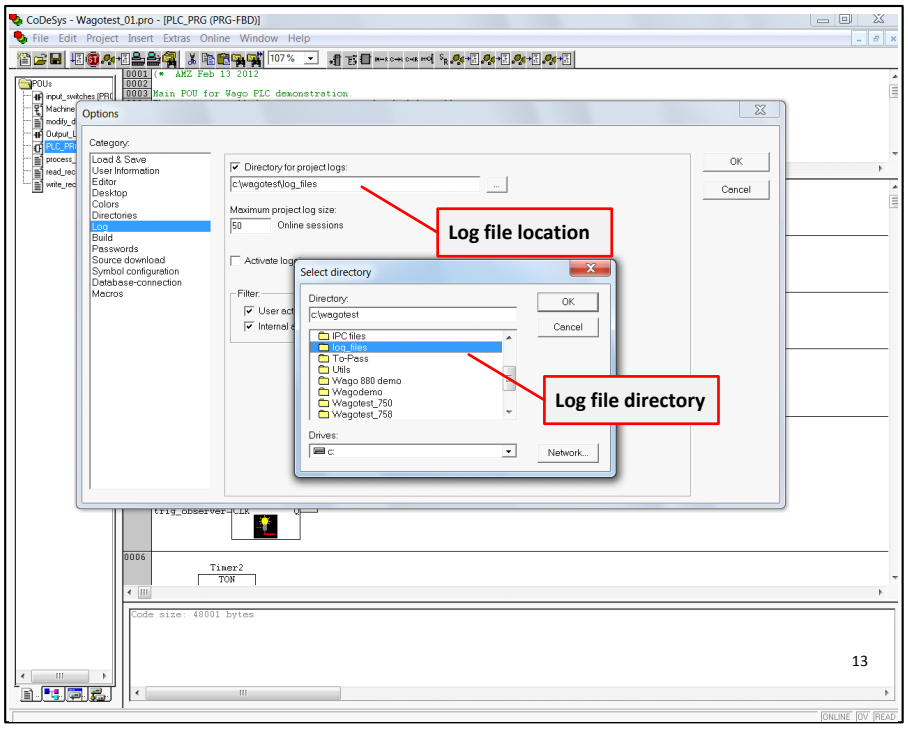
220. Claim 1 reads as follows:

> 1. An electronic audit system for an industrial control environment, comprising:

[1] a recording component to log real time interactions with one or more industrial control components; and

[2] a tracking component to aggregate the real time interactions to facilitate generation of audit data relating to the one or more industrial control components.

221. Claim 1 of the '974 patent is directed to "An electronic audit system for an industrial control environment, comprising:". I have been informed that this portion of the claim is called the preamble and does not give life and meaning to the claims, and therefore does not serve to limit the claim. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion that the WAGO-I/O System does provide an electronic audit system for an industrial control environment. The screenshot below depicts the "log" configuration screen in the Infringing Software that allows enabling of the audit data system including specification of a local or remote storage location for the associated audit data file. Additional reference information on this feature is also provided below.



(Testing of WAGO-I/O-PRO CAA)

6.5 Log

The log stores in chronological order actions that occur during an Online session. For this purpose a binary log file (*.log) is set up. Afterward, the user can store excerpts from the appropriate project log in an external log.

The log window can be opened in either Offline or Online mode and can thus serve as a direct monitor online.

'Window' 'Log'

To open, select the menu item 'Window' 'Log' or select entry 'Log' in the Resources tab.

In the log window, the filename of the currently displayed log appears after **Log:**. If this is the log of the current project, the word "(Internal)" will be displayed.

Registered entries are displayed in the log window. The newest entry always appears at the bottom.

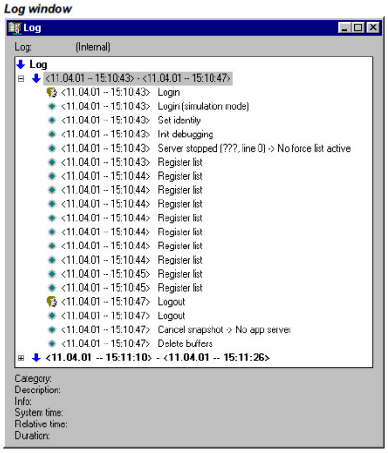
Only actions belonging to categories that have been activated in the 'Filter' field of the menu 'Project' 'Options' 'Log' will be displayed.

Available information concerning the currently selected entry is displayed below the log window:

Category: The category to which the particular log entry belongs. The following four categories are possible:

User action:	The user has carried out an Online action (typically from the Online menu).
Internal action:	An internal action has been executed in the Online layer (e.g. Delete Buffers or Init debugging).
Status change:	The status of the runtime system has changed (e.g. from Running to Break, if a breakpoint is reached).
Exception:	An exception has occurred, e.g. a communication error.

Log window



Description: The type of action. User actions have the same names as their corresponding menu commands; all other actions are in English and have the same name as the corresponding OnlineXXX() function.

Info: This field contains a description of an error that may have occurred during an action. The field is empty if no error has occurred.

System time: The system time at which the action began, to the nearest second.

Relative time: The time measured from the beginning of the Online session, to the nearest millisecond.

Duration: Duration of the action in milliseconds.

Menu Log

When the log window has the input focus, the menu option **Log** appears in the menu bar instead of the items 'Extras' and 'Options'.

The menu includes the following items:

Load... An external log file *.log can be loaded and displayed using the standard file open dialog.

The log that is present in the project will not be overwritten by the command. If the log window is closed and later opened again, or a new Online session is started then the version that is loaded will again be replaced by the project log.

(CoDeSys 2.3 Manual, p. 6-18-19)

(i) Element [1] of Claim 1

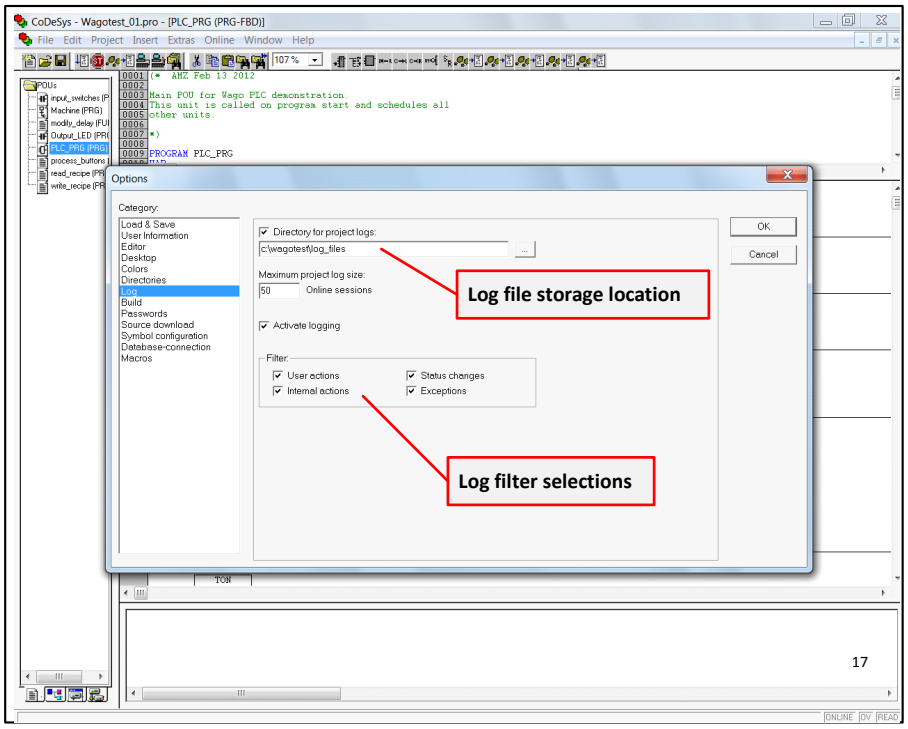
222. Element [1] of Claim 1 includes "a recording component to log real time interactions with one or more industrial control components." The Accused Instrumentalities contain a recording component to log real time interactions with one or more industrial control components.

223. Specifically, element [1] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. WAGO markets and sells the Infringing Software for use with Infringing PLCs. (See Decramer 30(b)(6) Dep. Tr. 24:5-26:9; see also Albers 30(b)(6) Dep. Tr. 46:11-23, 29:22-25, 34:13-15; WAGO-I/O-PRO CAA Compatibility Chart [ROCK0007738-ROCK0007744]; CoDeSys 2.3 Manual [ROCK0007278-ROCK0007737]; 750-841 Manual Section 3.1.7 [ROCK0008463]; 758-870/000-110 Manual Section 11-12 [ROCK0008687-ROCK0008720]).

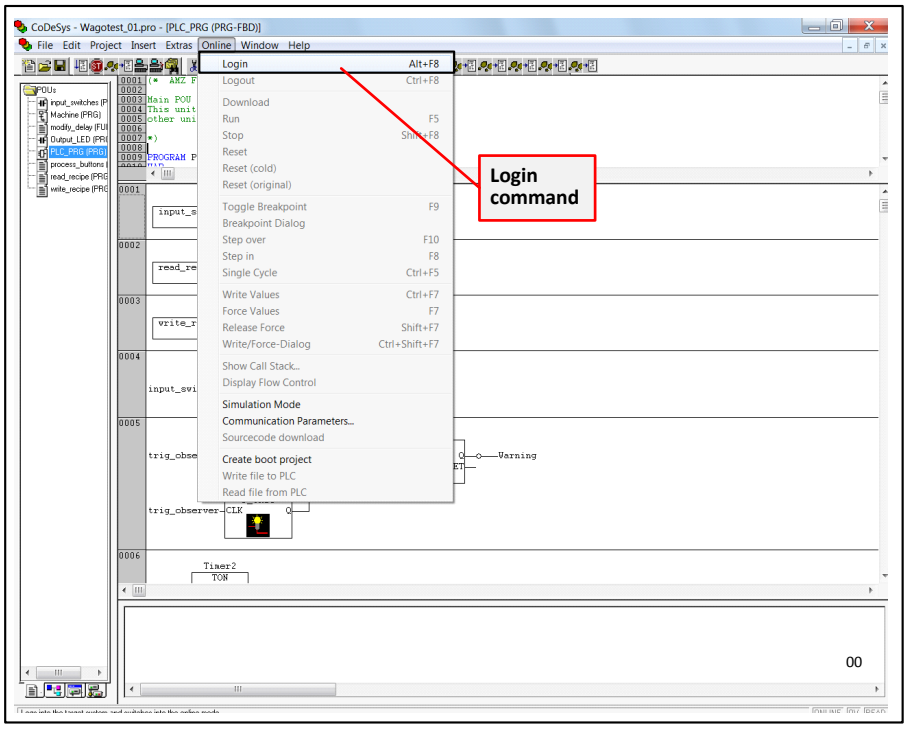
b. The Infringing Software records interactions in real time. (See Albers 30(b)(6) Dep. Tr. 100:3-14). This was observed in testing.

c. In the screenshot below, the Infringing Software can be configured to record real time interactions (user actions, status changes, internal actions, exceptions) with logging enabled through the listed menu options:



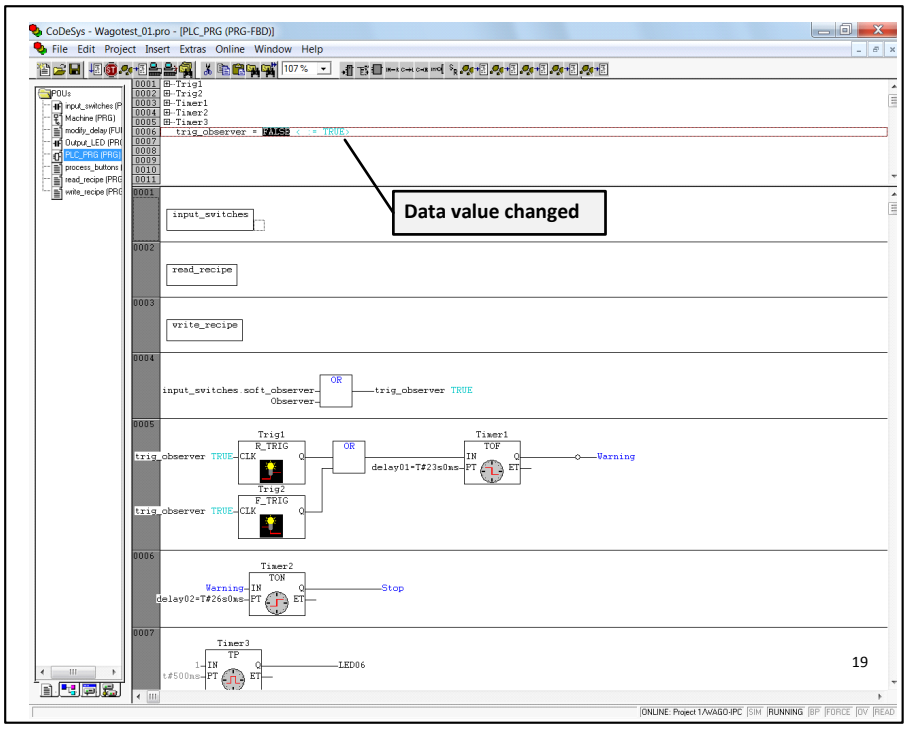
(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

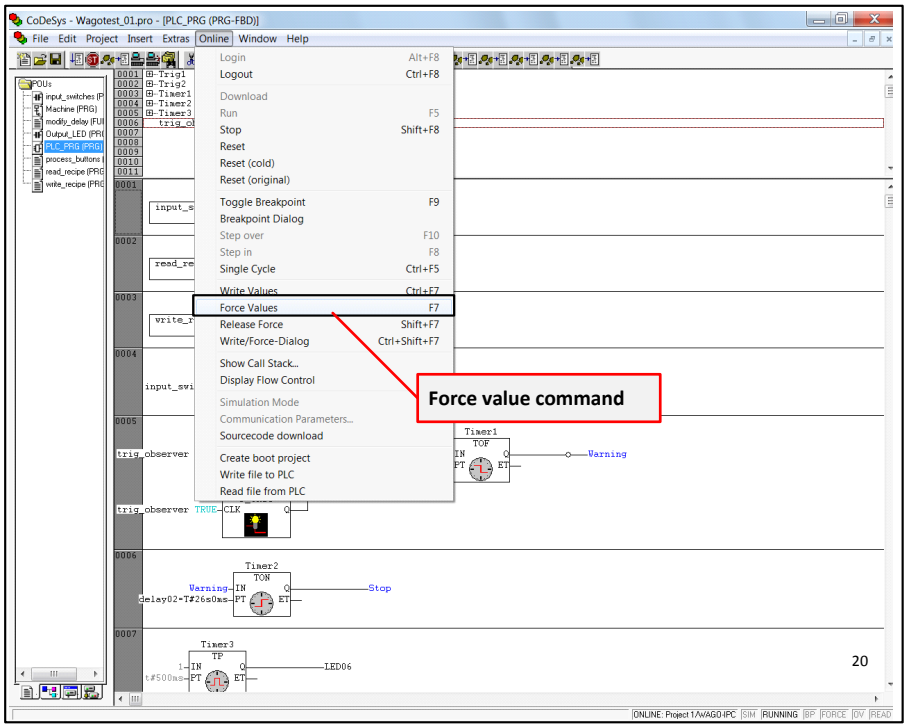
d. For illustration purposes, the value of "trig_observer" is "Forced" to be set to "TRUE" by performing the following steps, starting with logging into a running controller connected to the access tool:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller)

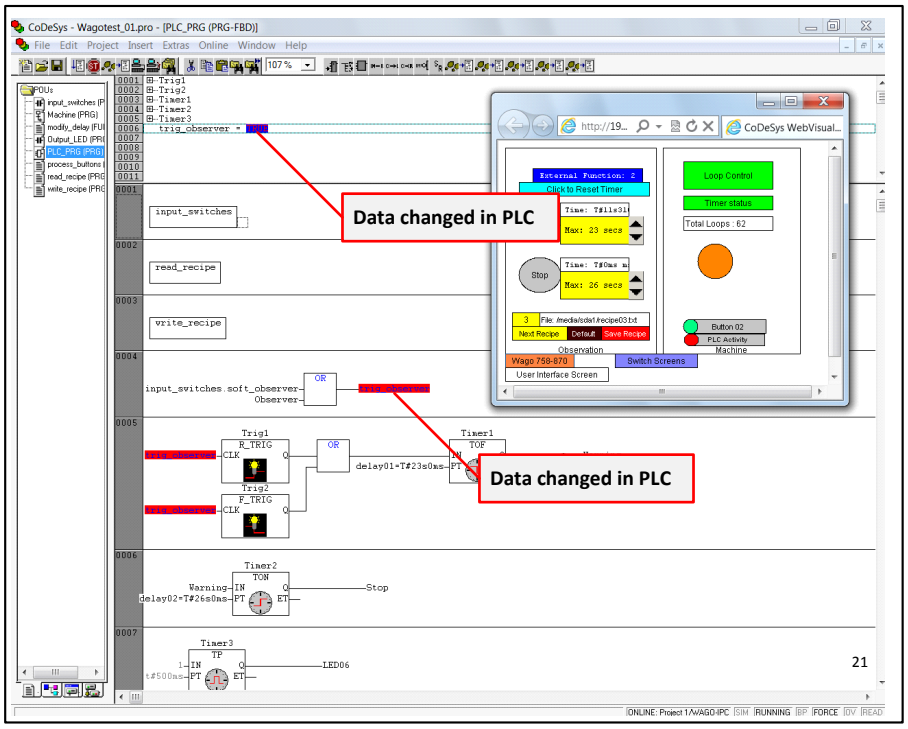
e. Next, a PLC data value is modified in the access tool, followed by use of a "force" action to cause a similar data change in the connected PLC:



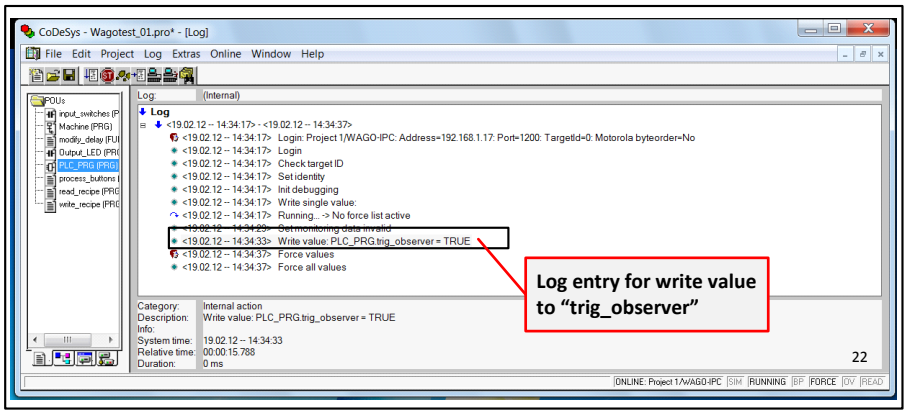


(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

f. In the screenshot below, the Infringing Software shows that the value of "trig_observer is "TRUE" in the controller after the "FORCE" action is executed.



The screen shot below demonstrates that the recording component recorded the interactions performed above.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities contain a recording component to log real time interactions with one or more industrial control

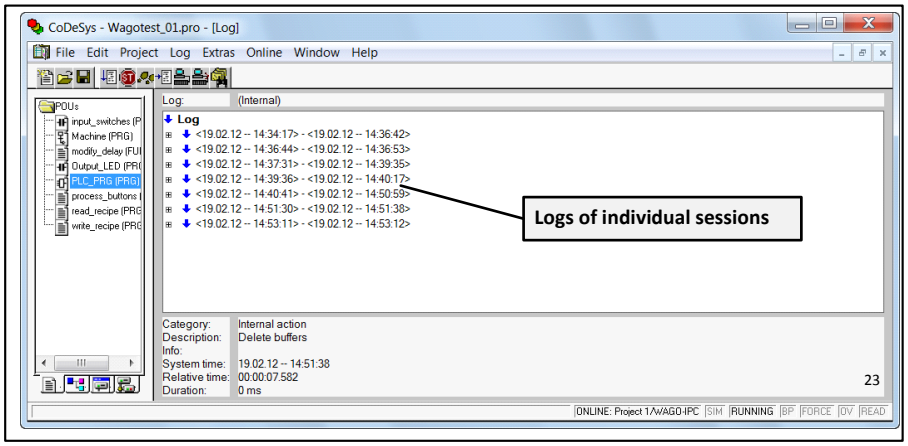
components as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(ii) Element [2] of Claim 1

224. Element [2] of Claim 1 includes, "a tracking component to aggregate the real time interactions to facilitate generation of audit data relating to the one or more industrial control components." The Accused Instrumentalities contain a tracking component to aggregate the real time interaction to facilitate generation of audit data relating to the one or more industrial control components.

225. Specifically, element [2] of Claim 1 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software tracks the interactions and aggregates in a log file. In the screenshot below the "log" file is opened and demonstrates that multiple sessions have been tracked and aggregated into sessions (denoted by time and date) with each session including the real time interactions for each session.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

> In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities contain a tracking component to aggregate the real time interaction to facilitate generation of audit data relating to the one or more industrial control components as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

226. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that the WAGO has infringed and is infringing claim 1 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

b. Claim 2

227. Claim 2 reads as follows:

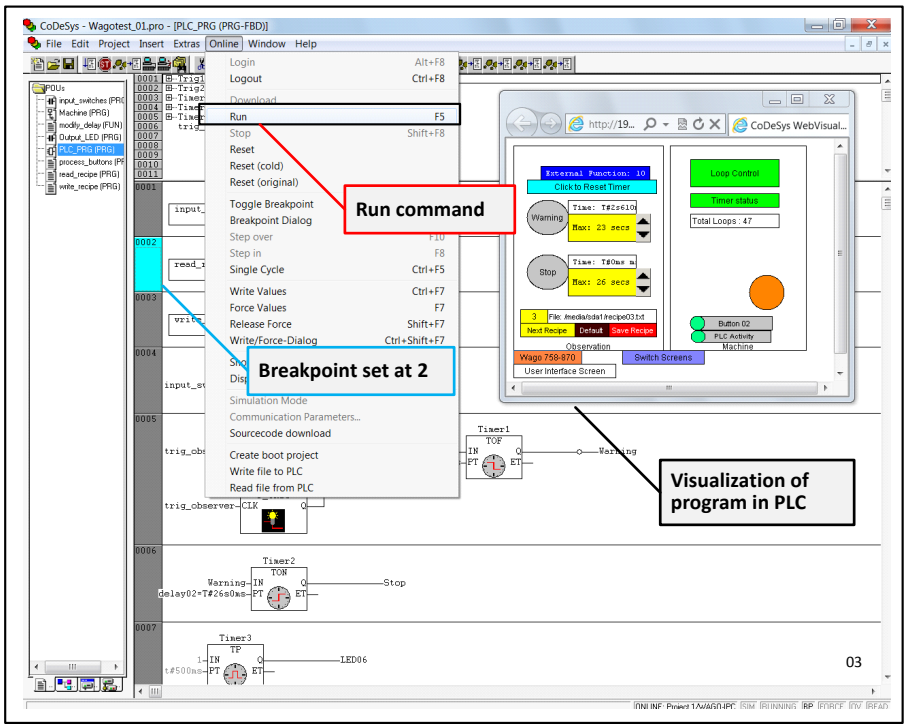
> 2. The system of claim 1, at least one of the recording component and the tracking component are associated with an access tool that interacts with the one or more industrial control components via a network.

228. Claim 2 includes the element "at least one of the recording component and the tracking component are associated with an access tool that interacts with the one or more industrial control components via a network." The Accused Instrumentalities provide an access tool that is associated with the recording component and/or the tracking component and that interacts with an Infringing PLC via a network.

229. Specifically, Claim 2 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

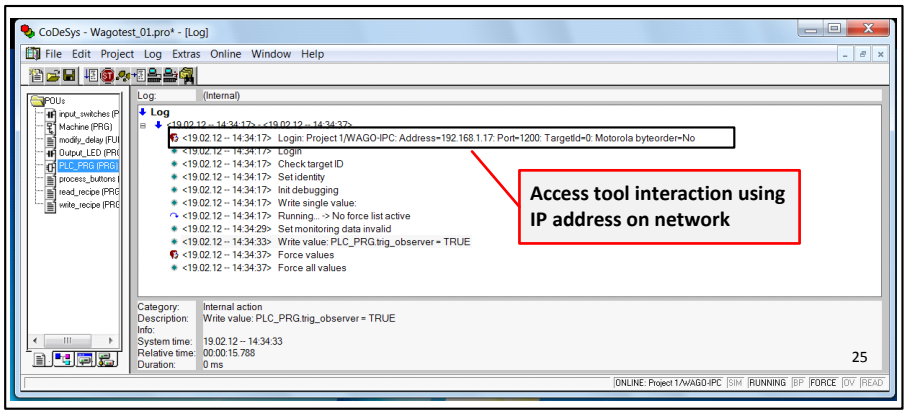
a. My testing confirmed that the Infringing Software communicated with an Infringing PLC via network (Ethernet link).

b. The Infringing Software has a built-in access tool that allows remote interaction via a network. The screenshot below shows the Infringing Software running in an "online mode" where the program is stored on an Infringing PLC:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

c. In the screenshot below, the recording and tracking components are associated with the access tool (Infringing Software) to record and track these interactions via a network (Ethernet link):



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities provide an access tool that is associated with the recording component and/or the tracking component and that interacts with an Infringing PLC via a network as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

230. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 2 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

c. <u>Claim 3</u>

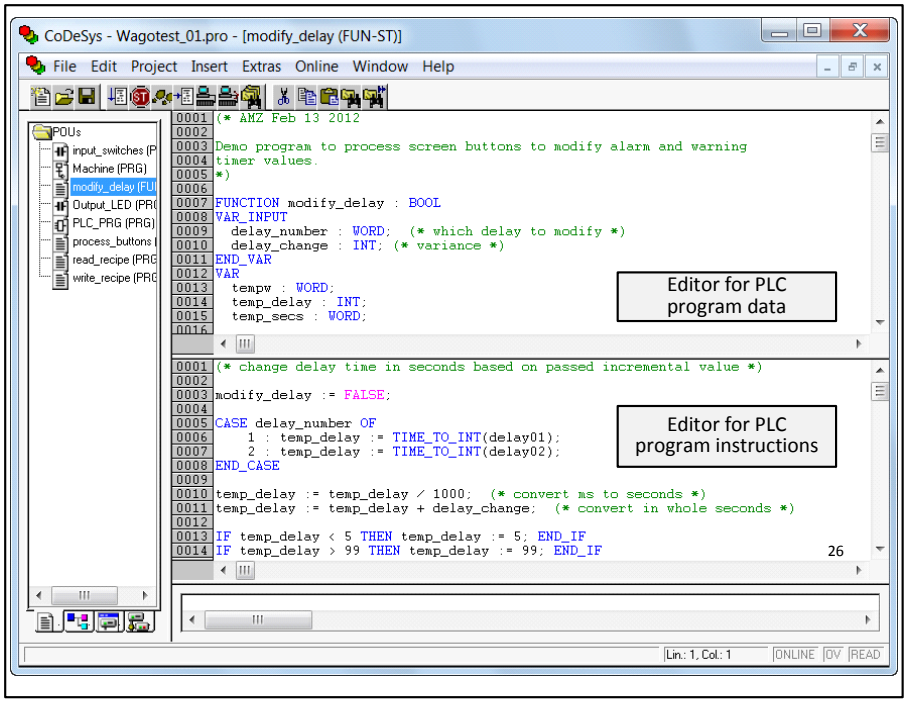
231. Claim 3 reads as follows:

> 3. The system of claim 2, the access tool includes at least one of an editing tool, a programming tool, a communications component, a monitoring component, a maintenance component, a browser, a graphical user interface (GUI), and a database application that interacts with the one or more industrial control components.

232. Claim 3 includes the element "the access tool includes at least one of an editing tool, a programming tool, a communications component, a monitoring component, a maintenance component, a browser, a graphical user interface (GUI), and a database application that interacts with the one or more industrial control components." The Accused Instrumentalities contain an access tool that includes at least one of an editing tool, a programming tool, a communications component, a monitoring component, a maintenance component, a browser, a graphical user interface (GUI) and a database application that interacts with the one or more industrial components.

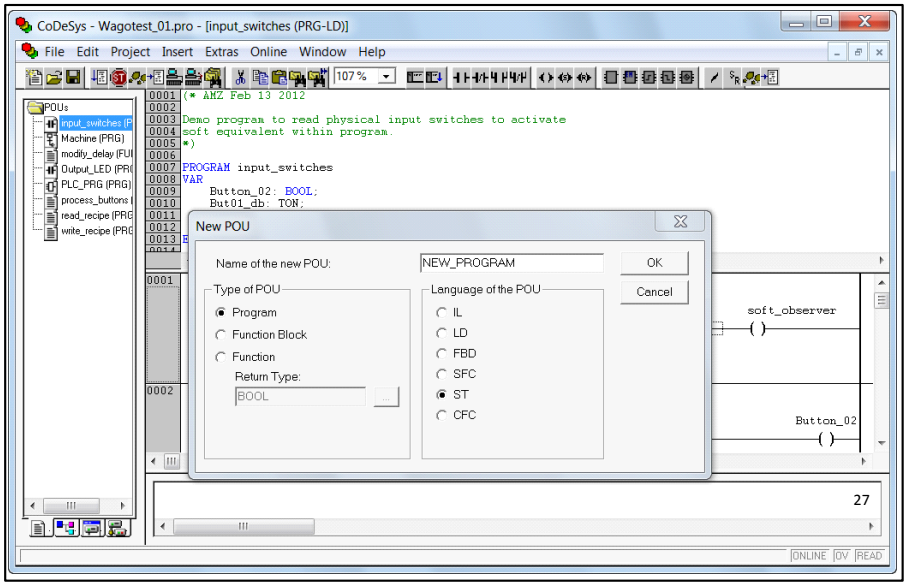
233. Specifically, Claim 3 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing Software for at least the following reasons:

a. The Infringing Software includes an editing tool. (See Albers 30(b)(6) Dep. Tr. 84:8-12).



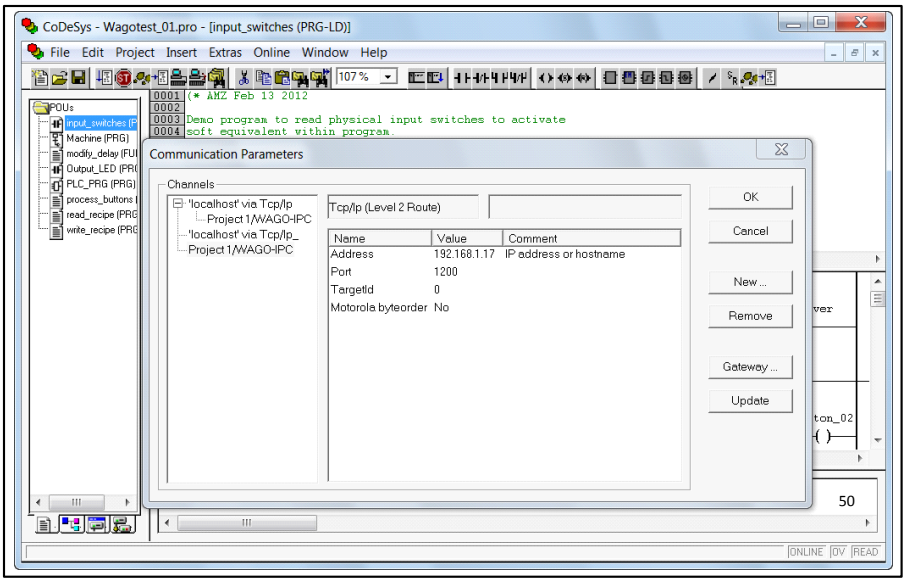
(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

b. The Infringing Software includes a programming tool that supports writing control programs in instruction list (IL), ladder diagram (LD), function block diagram (FBD), sequential function chart (SFC), structured text (ST) and continuous function chart (CFC). (See Albers 30(b)(6) Dep. Tr. 77:21-78:4, 84:8-12).



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

c. The Infringing Software includes a communications component.



d. The Infringing Software includes a monitoring component and has the ability to monitor the execution of a program. (See CoDeSys 2.3 Manual 2-24; 6-69).

Monitoring

In Online mode, all displayable variables are read from the controller and displayed in real time. You will find this display in the declarations and program editor; you can also read out current values of variables in the watch and receipt manager and in a visualization. If variables from instances of function blocks are to be monitored, the corresponding instance must first be opened.

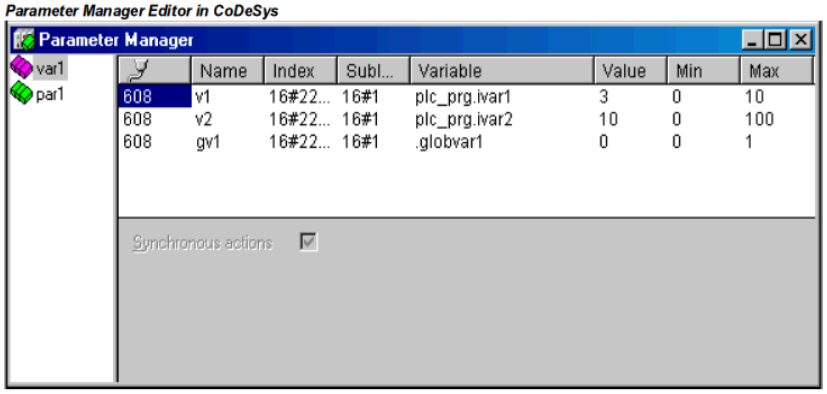
In monitoring VAR_IN_OUT variables, the de-referenced value is output.

In monitoring pointers, both the pointer and the de-referenced value are output in the declaration portion. In the program portion, only the pointer is output:

+ –pointervar = '<'pointervalue'>'

POINTERs in the de-referenced value are also displayed accordingly. With a simple click on the cross or a double-click on the line, the display is either expanded or truncated.

Parameter Manager Editor in CoDeSys



In online mode you can load the lists, you have created before, to the currently connected target system. You also can use the Parameter Manager functions to access them there for the purpose of data exchange with other systems (write values, upload). Further on in the Parameter Manager window you can monitor the current values of the parameters. If currently no online connection is established, the parameter lists just can be created locally and saved with the project.

e. The Infringing Software includes a maintenance component, for example forcing a value, which was confirmed by my test results.

f. The Infringing Software includes a GUI, as illustrated in the above screenshots.

g. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of

equivalents since the Accused Instrumentalities contain an access tool that includes at least one of an editing tool, a programming tool, a communications component, a monitoring component, a maintenance component, a browser, a graphical user interface (GUI) and a database application that interacts with the one or more industrial components as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

234. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 3 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

d. Claim 5

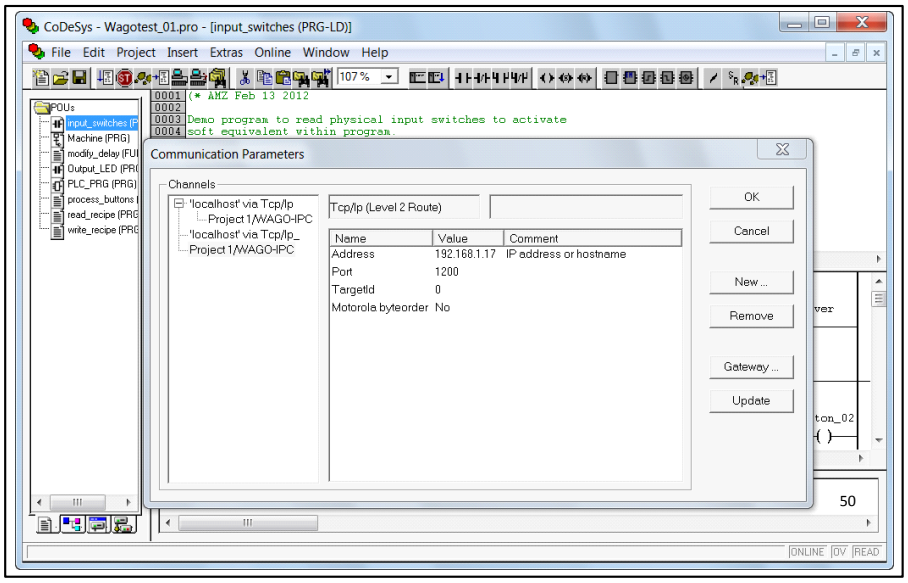
235. Claim 5 reads as follows:

> 5. The system of claim 2, the network includes at least one of a local factory network, a wireless network, and a public network.

236. Claim 5 includes the element "the network includes at least one of a local factory network, a wireless network, and a public network." The network used by the Accused Instrumentalities is a local factory network, a wireless network, or a public network.

237. Specifically, Claim 5 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software communicates using network protocols (e.g., TCP/IP) of the type typically used in local factory networks, wireless networks, and public networks:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

(See also Albers 30(b)(6) Dep. Tr. 107:20-108:4).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the network used by the Accused Instrumentalities is a local factory network, a wireless network, or a public network as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

238. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 5 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

e. <u>Claim 6</u>

239. Claim 6 reads as follows:

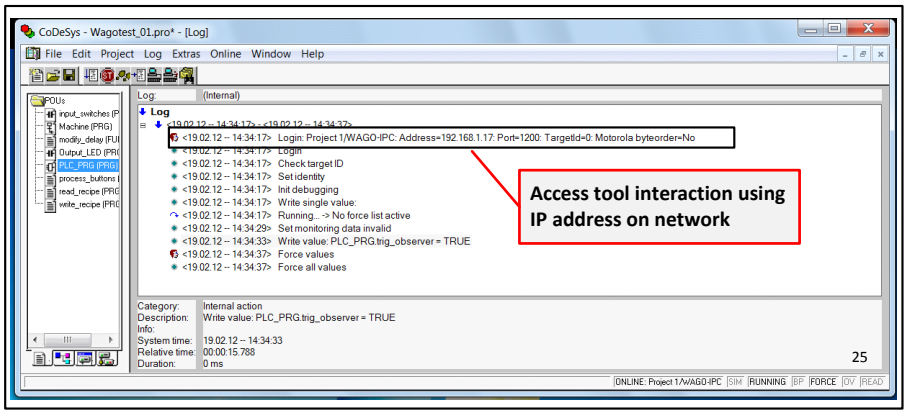
> 6. The system of claim 2, the recording component logs interaction data that has been directed to the one or more industrial control

components during a current application session associated with the access tool.

240. Claim 6 includes the element "the recording component logs interaction data that has been directed to the one or more industrial control components during a current application session associated with the access tool." The Accused Instrumentalities include a recording component that logs interaction data that has been directed to the one or more industrial control components during a current application session associated with the access tool.

241. Specifically, Claim 6 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The screen shot below demonstrates that the recording component logged interaction data with one industrial control component.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities include a recording component that logs interaction data that has been directed to the one or more industrial control components during a current application session

associated with the access tool as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

242. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 6 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

f. Claim 9

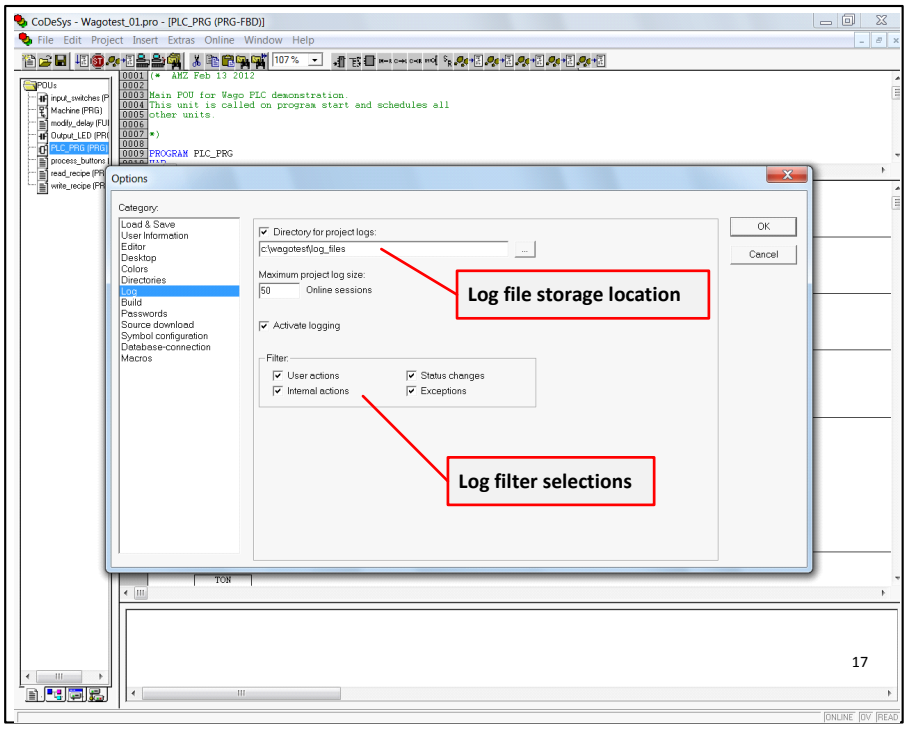
243. Claim 9 reads as follows:

> 9. The system of claim 1, the tracking component aggregates activities logged by the recording component in at least one of a local storage and a remote storage location.

244. Claim 9 includes the element "the tracking component aggregates activities logged by the recording component in at least one of a local storage and a remote storage location." The activities aggregated by the Accused Instrumentalities can be stored locally or in a remote storage location.

245. Specifically, Claim 9 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software can store the aggregated log data in a remote or a local storage location by selection the directory for project logs.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

b. The Infringing Software can store the aggregated log data remotely via USB memory stick or a networked drive. My testing has confirmed this result.

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the activities aggregated by the Accused Instrumentalities can be stored locally or in a remote storage location as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

246. Based on the foregoing, and based on other information disclosed in this report, it is my opinion WAGO has infringed and is infringing claim 9 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

g. Claim 10

247. Claim 10 reads as follows:

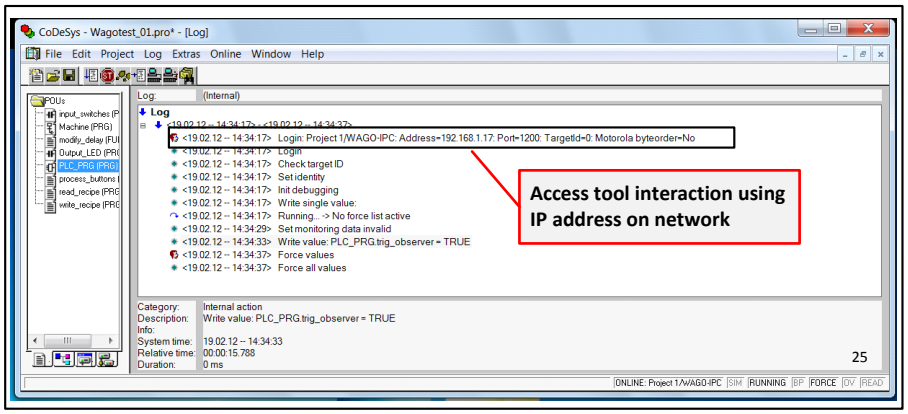
> 10. The system of claim 9, the tracking component aggregates transaction data by creating at least one of a file, schema, and a data structure in the local or remote storage locations, and tags the file, schema, and data structure with an identifier relating to the one or more industrial control components that have been accessed.

248. Claim 10 includes the element "the tracking component aggregates transaction data by creating at least one of a file, schema, and a data structure in the local or remote storage locations, and tags the file, schema, and data structure with an identifier relating to the one or more industrial control components that have been accessed." The transaction data is aggregated by the Accused Instrumentalities by creating at least one of a file, schema, and a data structure in the local or remote storage locations, and the file, schema, and data structure is tagged with an identifier relating to the one or more industrial control components that have been accessed.

249. Specifically, Claim 10 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software has the ability to save the aggregated data into a log file in a specified location as shown in the previous screenshot.

b. The log file contains a data structure that is tagged with an identifier relating to the industrial control component that has been accessed.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the transaction data is aggregated by the Accused Instrumentalities by creating at least one of a file, schema, and a data structure in the local or remote storage locations, and the file, schema, and data structure is tagged with an identifier relating to the one or more industrial control components that have been accessed as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

250. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 10 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

h. Claim 14

251. Claim 14 reads as follows:

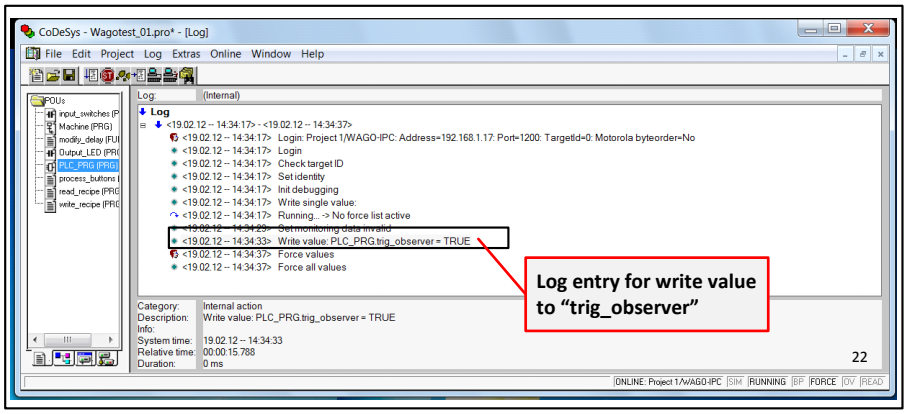
> 14. The system of claim 1, at least one of the recording component and the tracking component are employed to generate an audit

report that describes interactions that have occurred with the one or more industrial control components.

252. Claim 14 includes the element "at least one of the recording component and the tracking component are employed to generate an audit report that describes interactions that have occurred with the one or more industrial control components." The Accused Instrumentalities employ the recording component or the tracking component to generate an audit report that describes interactions that have occurred with the one or more industrial control components.

253. Specifically, Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software generates an audit report that can be viewed by users when the plus sign is clicked next to the log entry, which expands to display the recorded and tracked log data. This was confirmed by my testing. The report can also be printed, which was also confirmed by my testing:



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of

equivalents since the Accused Instrumentalities employ the recording component or the tracking component to generate an audit report that describes interactions that have occurred with the one or more industrial control components as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

254. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 14 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

i. Claim 16

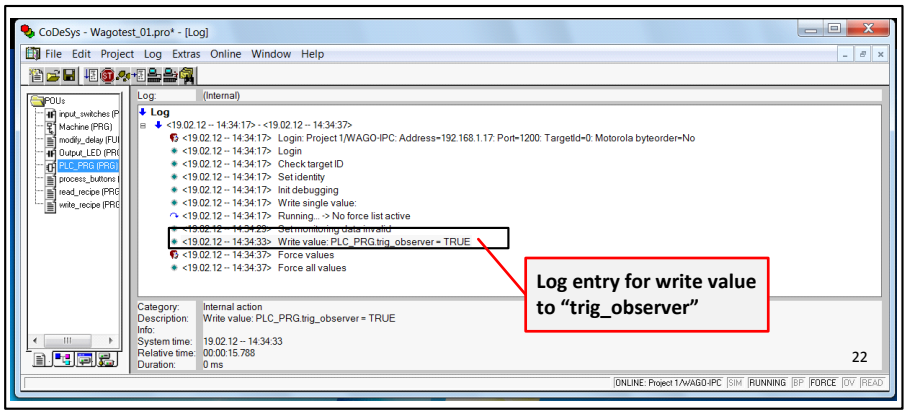
255. Claim 16 reads as follows:

> 16. The system of claim 14, the audit report includes 1 to N fields, N being an integer, the fields displaying various types of auditing information.

256. Claim 16 includes the element "the audit report includes 1 to N fields, N being an integer, the fields displaying various types of auditing information." The Accused Instrumentalities' audit report includes 1 or more fields displaying various types of auditing information.

257. Specifically, Claim 16 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. The Infringing Software displays audit report contains one or more fields containing various types of auditing information, including date, time, type of interaction, PLC data identifier, etc.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities' audit report includes 1 or more fields displaying various types of auditing information as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

258. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 16 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

j. <u>Claim 24</u>

259. Claim 24 reads as follows:

24. A method for verifying an industrial control process, comprising:

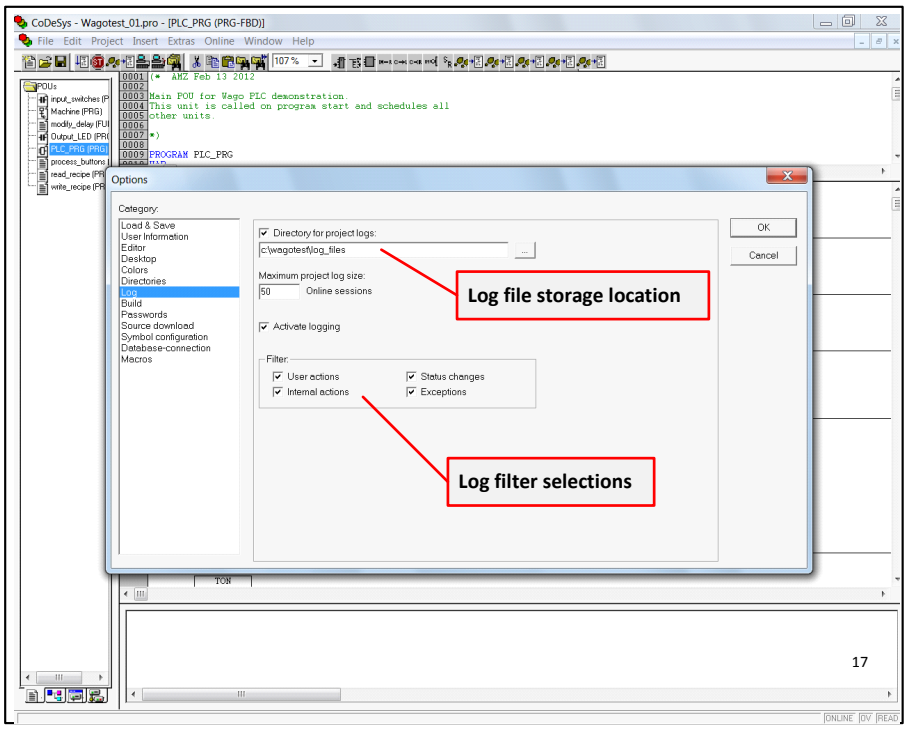
[1] monitoring activity data directed to one or more control components;

[2] tagging at least one file that is related to the or more control components;

[3] logging the activity data in at least one of a local and a remote location; and

[4] aggregating the logged activity data in the at least one file.

260. Claim 24 of the '974 patent is directed to "A method for verifying an industrial control process, comprising". I have been informed that this portion of the claim is called the preamble and does not give life and meaning to the claims, and therefore does not serve to limit the claims. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion that WAGO does practice a method for verifying an industrial control process. The screenshot below depicts the "log" configuration screen in the Infringing Software that allows enabling of the audit data system including specification of a local or remote storage location for the associated audit data file. Additional reference information on this feature is also provided below.:



(Testing of CoDeSys 2.3).

6.5 Log

The log stores in chronological order actions that occur during an Online session. For this purpose a binary log file (*.log) is set up. Afterward, the user can store excerpts from the appropriate project log in an external log.

The log window can be opened in either Offline or Online mode and can thus serve as a direct monitor online.

'Window' 'Log'

To open, select the menu item 'Window' 'Log' or select entry 'Log' in the Resources tab.

In the log window, the filename of the currently displayed log appears after **Log:**. If this is the log of the current project, the word "(Internal)" will be displayed.

Registered entries are displayed in the log window. The newest entry always appears at the bottom.

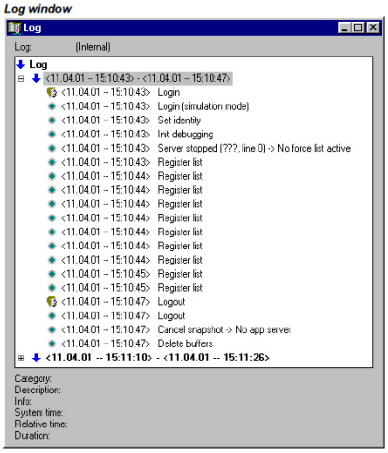
Only actions belonging to categories that have been activated in the 'Filter' field of the menu 'Project' 'Options' 'Log' will be displayed.

Available information concerning the currently selected entry is displayed below the log window:

Category: The category to which the particular log entry belongs. The following four categories are possible:

User action:	The user has carried out an Online action (typically from the Online menu).
Internal action:	An internal action has been executed in the Online layer (e.g. Delete Buffers or Init debugging).
Status change:	The status of the runtime system has changed (e.g. from Running to Break, if a breakpoint is reached).
Exception:	An exception has occurred, e.g. a communication error.

Log window



Description: The type of action. User actions have the same names as their corresponding menu commands; all other actions are in English and have the same name as the corresponding OnlineXXX() function.

Info: This field contains a description of an error that may have occurred during an action. The field is empty if no error has occurred.

System time: The system time at which the action began, to the nearest second.

Relative time: The time measured from the beginning of the Online session, to the nearest millisecond.

Duration: Duration of the action in milliseconds.

Menu Log

When the log window has the input focus, the menu option **Log** appears in the menu bar instead of the items 'Extras' and 'Options'.

The menu includes the following items:

Load... An external log file *.log can be loaded and displayed using the standard file open dialog.

The log that is present in the project will not be overwritten by the command. If the log window is closed and later opened again, or a new Online session is started then the version that is loaded will again be replaced by the project log.

(CoDeSys 2.3 Manual, p. 6-18-19)

(i) Element [1] of Claim 24

261. Element [1] of Claim 24 includes the element "monitoring activity data directed to one or more control components;" The Accused Instrumentalities monitor activity data directed to one or more control components.

262. Specifically, element [1] of Claim 24 is met literally, or in the alternative, under the doctrine of equivalents by the Accused Instrumentalities for at least the following reasons:

a. WAGO markets and sells the Infringing Software for use with Infringing PLCs.

b. The Infringing Software monitors interactions with a control component:

Monitoring

In Online mode, all displayable variables are read from the controller and displayed in real time. You will find this display in the declarations and program editor; you can also read out current values of variables in the watch and receipt manager and in a visualization. If variables from instances of function blocks are to be monitored, the corresponding instance must first be opened.

In monitoring VAR_IN_OUT variables, the de-referenced value is output.

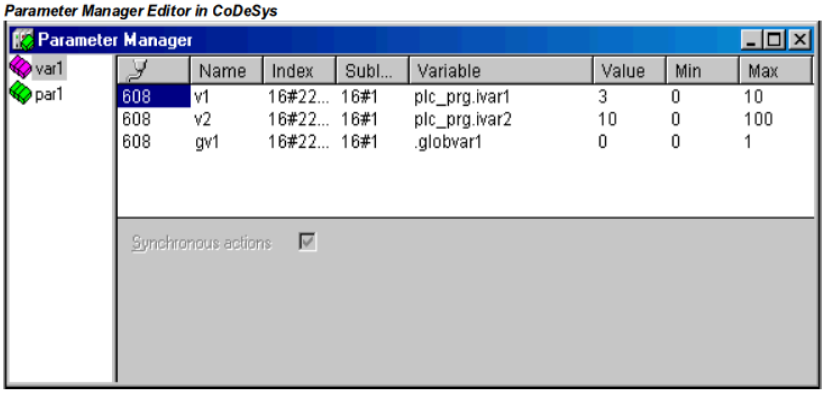
In monitoring pointers, both the pointer and the de-referenced value are output in the declaration portion. In the program portion, only the pointer is output:

+ –pointervar = '<'pointervalue'>'

POINTERs in the de-referenced value are also displayed accordingly. With a simple click on the cross or a double-click on the line, the display is either expanded or truncated.

See CoDeSys 2.3 Manual 2-24

Parameter Manager Editor in CoDeSys



In online mode you can load the lists, you have created before, to the currently connected target system. You also can use the Parameter Manager functions to access them there for the purpose of data exchange with other systems (write values, upload). Further on in the Parameter Manager window you can monitor the current values of the parameters. If currently no online connection is established, the parameter lists just can be created locally and saved with the project.

See CoDeSys 2.3 Manual 6-69.

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Accused Instrumentalities monitor activity data directed to one or more control components as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(ii) Element [2] of Claim 24

263. Element [2] of Claim 24 includes the element "tagging at least one file that is related to the or more control components." The Infringing Software tags at least one file that is related to the industrial control.

264. Specifically, element [2] of Claim 24 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing Software for at least the following reasons:

a. The Infringing Software creates and tags a file to store audit data from interactions with the industrial controller. My testing confirms this result.

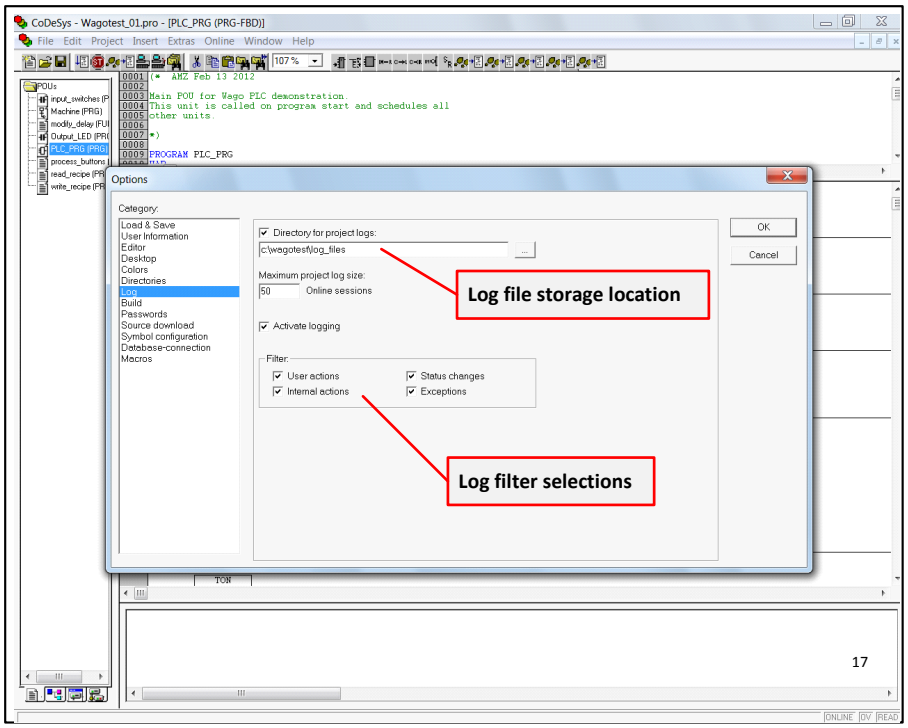
b. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing Software tags at least one file that is related to the industrial control as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(iii) Element [3] of Claim 24

265. Element [3] of Claim 24 includes the element "logging the activity data in at least one of a local and a remote location." The Infringing Software logs the activity data either locally or in a remote location.

266. Specifically, element [3] of Claim 24 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing Software for at least the following reasons:

a. The Infringing Software logs the interactions in a remote or a local storage location by selection of the directory for project logs.



(Testing of CoDeSys 2.3 with WAGO 758-870 Controller).

b. The Infringing Software can log data remotely via USB memory stick or a networked drive. My testing has confirmed this result.

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of

equivalents since the Infringing Software logs the activity data either locally or in a remote location as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(iv) Element [4] of Claim 24

267. Element [4] of Claim 24 includes the element "aggregating the logged activity data in the at least one file." The Infringing Software aggregates the logged activity data in a file.

268. Specifically, element [4] of Claim 24 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing Software for at least the following reasons:

a. The Infringing Software saves the aggregated data into the tagged file as shown above. My testing confirms this result.

b. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing Software aggregates the logged activity data in a file as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

269. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 24 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

270. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are

using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

271. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, these components have no substantial, non-infringing uses, and the components constitute a material part of Rockwell's invention. WAGO is aware of the '974 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '974 patent.

272. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 24 of the '974 patent.

273. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 24 of the '974 patent.

o. Claim 29

274. Claim 29 reads as follows:

> 29. A computer readable medium having stored thereon a data structure, comprising:
>
> [1] a first data field representing real time access data to an industrial control component;
>
> [2] a second data field representing a tag name to store and aggregate the real time access data; and
>
> [3] a third data field to categorize the real time access data.

275. To a PHOSITA, "a computer readable medium" refers to non-volatile memory used to store and retrieve data or instructions for computer processors.

276. The Infringing Software provides [1] a first data field representing real time access data to an industrial control component; [2] a second data field representing a tag name to store and aggregate the real time access data; and [3] a third data field to categorize the real time access data. See infringement analysis above for the prior claims in the '974 patent.

277. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing Software provides [1] a first data field representing real time access data to an industrial control component; [2] a second data field representing a tag name to store and aggregate the real time access data; and [3] a third data field to categorize the real time access data as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

278. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 29 of the '974 patent literally, or in the alternative, under the doctrine of equivalents.

3. WAGO's Infringement of U.S. Patent No. 6,801,813

279. I understand that Plaintiffs are asserting claims 1-7, 10-17, and 20-22 of the '813 patent. It is my opinion that WAGO has been and is infringing claims 1-7, 10-17, and 20-22 of the '813 patent, has induced and is inducing infringement of claims 21 and 22 of the '813 patent, and has contributed to their customers' infringement of claims 21 and 22 of the '813 patent, either literally or under the doctrine of equivalents.

280. Based on my review, I believe that the Accused Instrumentalities practice each and every element of claims 1-7, 10-17, and 20-22 of the '813 patent. My opinion is based on my tests, my experience, the documents I reviewed and other information in this report.

281. I do not believe any of the claim terms in the '813 patent require explanation or a Court's interpretation. The terms are clear and easily understood not only by a PHOSITA, but simply by laypeople as well. It is my opinion that the terms of the '813 patent are understood in light of the patent's specification and the claims themselves, such that no construction should be adopted that alters the plain meaning of these words.

282. If the Court construes the term "recipe file" as used in the '813 patent, it should mean "a profile and associated parameters regarding a particular implementation or process." This definition is supported by my experience in the field, and the intrinsic evidence in '813 patent and its prosecution history.

283. While the term "editor for developing ladder logic programs" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "editor for developing ladder logic programs," it is my opinion that the claim elements are satisfied for infringement.

284. While the term "execution engine" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the

Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. Defendants' definition is not supported by the intrinsic evidence of the '813 patent. Defendants attempt to limit "execution engine" to an interpreter which is in my opinion too narrow of a construction based on the way the term is used in the '813 patent, and also based on my experience with compiled and interpreted computer processing.

285. While the term "file system" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "file system," it is my opinion that the claim elements are satisfied for infringement.

286. While the term "file system services" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "file system services," it is my opinion that the claim elements are satisfied for infringement.

287. While the term "header instructions" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "header instructions," it is my opinion that the claim elements are satisfied for infringement.

288. While the term "industrial control program" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "industrial control program," it is my opinion that the claim elements are satisfied for infringement.

289. While the term "industrial controller" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. Moreover, Plaintiffs' position supports the notion that an "industrial controller" in the context of the patent is a PLC. However, taking either

party's construction of the term "industrial controller," it is my opinion that the claim elements are satisfied for infringement.

290. While the term "ladder logic" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "ladder logic," it is my opinion that the claim elements are satisfied for infringement.

291. While the term "load instruction" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "load instruction," it is my opinion that the claim elements are satisfied for infringement.

292. While the term "program memory" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. However, taking either party's construction of the term "program memory," it is my opinion that the claim elements are satisfied for infringement.

a. Claim 1

293. Claim 1 reads as follows:

> 1. An industrial controller system comprising: a file system residing in a program memory of an industrial controller, the file system having a plurality of file system services; and an execution engine residing in the program memory of the industrial controller, the execution engine adapted to interpret code from an industrial control program, the industrial control program including at least one instruction utilizing one or more of the plurality of file system services.

294. Claim 1 of the '813 patent is directed to "An industrial controller system comprising". I have been informed that this portion of the claim is called the preamble and does not give life and meaning to the claims, and therefore does not serve to limit the claims. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion WAGO makes, offers to sell, sells, and imports an industrial controller system known as the WAGO-I/O System. (See Full Line Catalog 2010/2011).

295. Claim 1 includes the element "a file system residing in a program memory of an industrial controller, the file system having a plurality of file system services; and an execution engine residing in the program memory of the industrial controller, the execution engine adapted to interpret code from an industrial control program, the industrial control program including at least one instruction utilizing one or more of the plurality of file system services." The Infringing File System PLCs are industrial controllers or PLCs with a file system residing in a program memory, and the file system has a plurality of file system services. The Infringing File System PLCs also have an execution engine adapted to interpret code from an industrial control program that has at least one instruction utilizing one or more of the plurality of file system services of the PLC.

296. Specifically, Claim 1 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

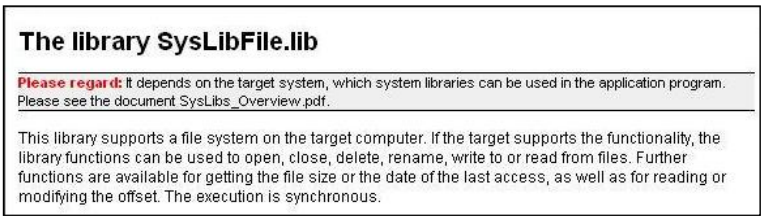
a. The Infringing File System PLCs comprise PLCs such as the WAGO Programmable Field Bus Controllers (750 PLCs) and WAGO IPC products (758 PLCs) that contain file systems residing in the program memory of those controllers. (See www.wago.us/products/37712.htm; Albers 30(b)(6) Dep. Tr. 71:4-15; 71:25-72:5).

b. The first WAGO-I/O System PLC that had an operating system was the 750-841. (See Albers 30(b)(6) Dep. Tr. 73:12-16).

c. I am informed that the WAGO 750-841 was first sold sometime in 2003. (See Albers 30(b)(6) Dep. Tr. 73:22-25).

d. WAGO-I/O System PLCs that contain an operating system, such as the 750-841, have a file system residing in the program memory of the PLC. (See Albers 30(b)(6) Dep. Tr. 75:8-12).

e. The file system on an Infringing File System PLC has a plurality of file system services, which can be utilized by the SysLibFile.lib function. (See "Using the SysLibFile.lib and the WagoLibFtp.lib for file access" Application Note Version 1.0.0 pp. 6-7, 10 [ROCK0007865-66, 7869]).

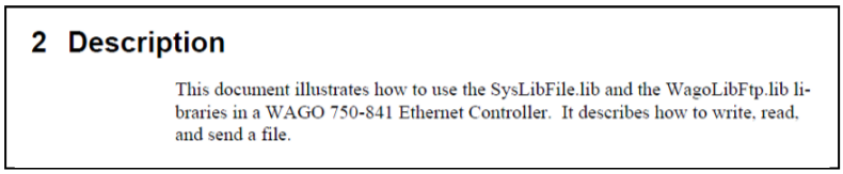

The library SysLibFile.lib

Please regard: It depends on the target system, which system libraries can be used in the application program. Please see the document SysLibs_Overview.pdf.

This library supports a file system on the target computer. If the target supports the functionality, the library functions can be used to open, close, delete, rename, write to or read from files. Further functions are available for getting the file size or the date of the last access, as well as for reading or modifying the offset. The execution is synchronous.

(CoDeSys v.2.3 program help; see also other libraries providing other services in WAGO and CoDeSys manuals)

h. Since WAGO has not provided the system development kit for the Infringing Software, I am unable to know exactly how the execution is implemented on the various Infringing File System PLCs. However, my testing shows that the Infringing File System PLCs use portions of the Infringing Software (including the CoDeSys runtime system) to execute an industrial control program in the PLC.



2 Description

This document illustrates how to use the SysLibFile.lib and the WagoLibFtp.lib libraries in a WAGO 750-841 Ethernet Controller. It describes how to write, read, and send a file.

(Wago Application note A114100 Page 6)

i. I am told that the runtime system is installed on the Infringing File System PLCs prior to the download of the control program. (See Albers 30(b)(6) Dep. Tr. at 86:5-15; see also CoDeSys Brochure WCP10006).

j. The runtime system resides in the program memory of the PLCs. (See Albers 30(b)(6) Dep. Tr. 78:23-79:80-12, 142:18-143:3).

k. The runtime system is adapted to interpret code from an industrial control program, the industrial control program including at least one instruction utilizing one or more of the plurality of file system services of the PLC. (See Albers 30(b)(6) Dep. Tr. 78:8-12, 80:12; 86:5-15; 142:7-12).

l. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of

equivalents since the Infringing File System PLCs are PLCs with a file system residing in a program memory, and the file system has a plurality of file system services as set forth herein. The Infringing File System PLCs also have an execution engine adapted to interpret code from an industrial control program that has at least one instruction utilizing one or more of the plurality of file system services of the PLC. As such, it is my view that the Infringing File System PLCs would perform substantially the same function in substantially the same way to yield substantially the same result, and thus infringe under the doctrine of equivalents.

297. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing and are infringing Claim 1 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

b. Claim 2

298. Claim 2 reads as follows:

> 2. The system of claim 1, the file system and the execution engine being adapted to load user defined routine files upon loading an industrial control program having one or more header instructions for including a user defined routine file, the included user defined routine file being loaded into the same program space as the industrial control program.

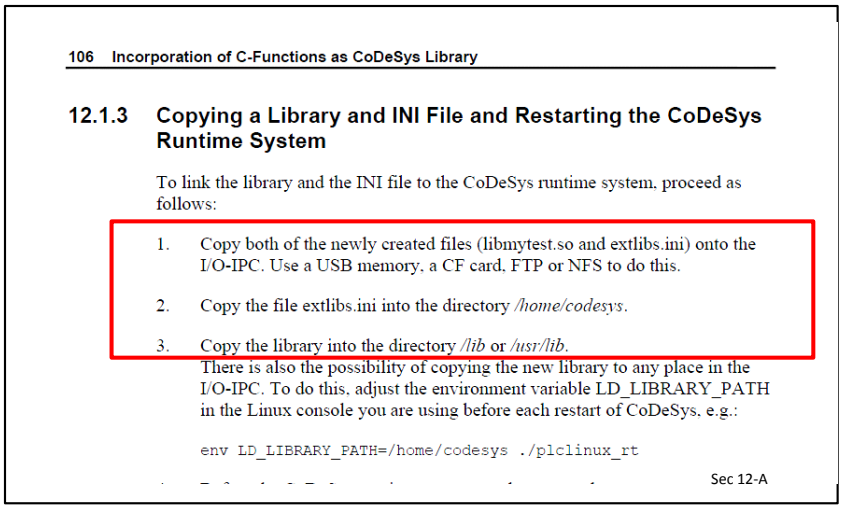
299. Claim 2 includes the element "the file system and the execution engine being adapted to load user defined routine files upon loading an industrial control program having one or more header instructions for including a user defined routine file, the included user defined routine file being loaded into the same program space as the industrial control program." On information and belief, the Infringing File System PLCs load user defined routine files upon loading an industrial control program having one or more header instructions for including a user

defined routine file, and the user defined routine file is loaded into the same program space as the industrial control program.

300. Specifically, Claim 2 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing WAGO-I/O System for at least the following reasons:

a. The file system and execution engine of the 758 PLCs are adapted to use user-defined external functions (i.e. user routines) that are loaded by the controller's execution engine at runtime. (See 758-870 Manual Section 12 [ROCK0008711-ROCK0008720]).

Implementing a user defined routine file in the Infringing Software includes generation of a header file (i.e. an ".h" file) on the development system to identify the routines that will be loaded by the CoDeSys runtime system on the PLC. The user defined routine is contained in "C-Function" external library file with type of ".so" that must be copied to the controller's file system. The implementation also requires that an ",ini" initialization file be created and copied to the CoDeSys directory on the controller. The ini file provides the header instructions to the CoDeSys execution engine to use file services to load the user defined routine from the library file when the control program is started on the PLC.



106 Incorporation of C-Functions as CoDeSys Library

12.1.3 Copying a Library and INI File and Restarting the CoDeSys Runtime System

To link the library and the INI file to the CoDeSys runtime system, proceed as follows:

1. Copy both of the newly created files (libmytest.so and extlibs.ini) onto the I/O-IPC. Use a USB memory, a CF card, FTP or NFS to do this.
2. Copy the file extlibs.ini into the directory */home/codesys*.
3. Copy the library into the directory */lib* or */usr/lib*.
   There is also the possibility of copying the new library to any place in the I/O-IPC. To do this, adjust the environment variable LD_LIBRARY_PATH in the Linux console you are using before each restart of CoDeSys, e.g.:

```
env LD_LIBRARY_PATH=/home/codesys ./plclinux_rt
```

Sec 12-A

b. The WAGO 758 PLC's execution engine runtime system loads and links the user-defined routines into the control program during program startup, and then executes the control program and user defined routines in the program memory on the controller. This entire user defined routine system was confirmed by my testing.

c. Since WAGO has not provided the system development kit for the Infringing Software, I am unable to know exactly how the 750 PLCs load user routine files based. However, I note that WAGO's 30(b)(6) witness responsible for testifying regarding the functionality of the Infringing File System PLCs stated that a program executing on an Infringing File System PLC can call an external function that is outside of the compiled executable code. (See Albers 30(b)(6) Dep. Tr. 80:25-81:3). Therefore I assume, the WAGO 750 PLCs call external functions in a similar fashion to the WAGO 758 PLCs.

d. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this

claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs load user defined routine files upon loading an industrial control program having one or more header instructions for including a user defined routine file, and the user defined routine file is loaded into the same program space as the industrial control programs as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

301. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 2 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

c. Claim 3

302. Claim 3 reads as follows:

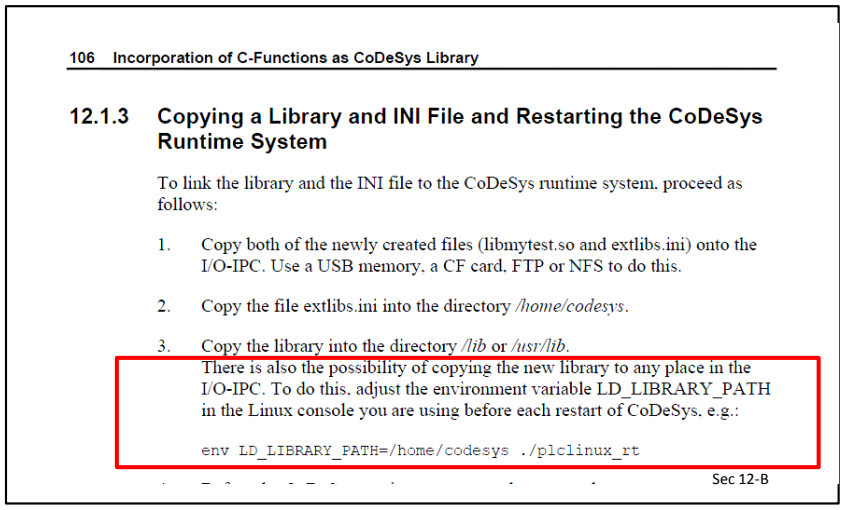
> 3. The system of claim 2, the user defined routine files being stored at a memory device separate from the program memory.

303. Claim 3 includes the element "the user defined routine files being stored at a memory device separate from the program memory." The Infringing File System PLCs can load user defined routine files that are stored separate from the program memory.

304. Specifically, Claim 3 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The execution engine of the Infringing PLCs can use the file services to load user defined routines from any storage location that is accessible to the controller's file system. (See 758-870 Manual Section 12

[ROCK0008711-ROCK0008720]). This file storage memory is separate from the program memory.



106 Incorporation of C-Functions as CoDeSys Library

12.1.3 Copying a Library and INI File and Restarting the CoDeSys Runtime System

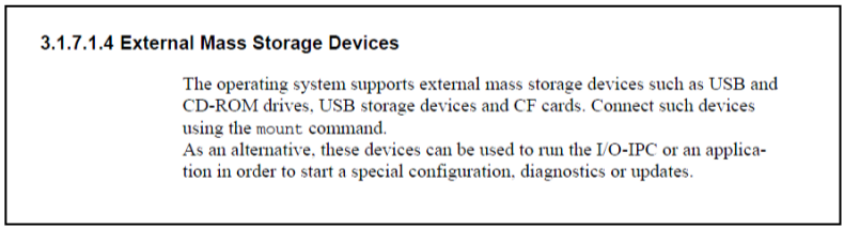
To link the library and the INI file to the CoDeSys runtime system, proceed as follows:

1. Copy both of the newly created files (libmytest.so and extlibs.ini) onto the I/O-IPC. Use a USB memory, a CF card, FTP or NFS to do this.
2. Copy the file extlibs.ini into the directory */home/codesys*.
3. Copy the library into the directory */lib* or */usr/lib*.
   There is also the possibility of copying the new library to any place in the I/O-IPC. To do this, adjust the environment variable LD_LIBRARY_PATH in the Linux console you are using before each restart of CoDeSys, e.g.:

   `env LD_LIBRARY_PATH=/home/codesys ./plclinux_rt`

Sec 12-B

b. The Infringing File System PLCs can access storage locations that are separate from the PLC's storage memory, such as USB memory sticks, Compact Flash, SD cards, external hard drives.



3.1.7.1.4 External Mass Storage Devices

The operating system supports external mass storage devices such as USB and CD-ROM drives, USB storage devices and CF cards. Connect such devices using the `mount` command.
As an alternative, these devices can be used to run the I/O-IPC or an application in order to start a special configuration, diagnostics or updates.

(2004 System 758 Manual, p. 63)

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs can load user defined routine files that are stored separate from the program memory as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

305. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 3 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

d. Claim 4

306. Claim 4 reads as follows:

> 4. The system of claim 3, the memory device being located at one of the industrial controller and a remote location from the industrial controller.

307. Claim 4 includes the element "the memory device being located at one of the industrial controller and a remote location from the industrial controller." The Infringing File System PLCs can load user defined routine files that are stored at least at the controller location.

308. Specifically, Claim 4 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The user defined routine files can be stored anywhere in the Infringing File System PLC's file system. (See 758-870 Manual Section 12 [ROCK0008711-ROCK0008720]).

b. The Infringing File System PLC can access a variety of local storage locations such as USB memory sticks, Compact Flash, SD cards, external hard drives.

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs can load user defined routine files that are stored at least at the controller location as set forth

herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

309. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 4 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

e. Claim 5

310. Claim 5 reads as follows:

> 5. The system of claim 1, the file system and the execution engine being adapted to load one or more recipe files into an executing industrial control program upon executing a load instruction in an industrial control program.

311. Claim 5 includes the element "the file system and the execution engine being adapted to load one or more recipe files into an executing industrial control program upon executing a load instruction in an industrial control program." The file system and the execution engine of the Infringing File System PLCs are adapted to load one or more recipe files into an executing industrial control program upon executing a load instruction in an industrial control program.

312. Specifically, Claim 5 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. WAGO's 30(b)(6) witness testified that a recipe is a data set on how a certain machine has to be parametered or what parameters a machine has to have in order to get a specific result. (Albers 30(b)(6) 76:10-18).

b. A program executing in the Infringing File System PLCs can load a recipe file stored in the file system. (Albers 30(b)(6) Dep. Tr. 92:24-93:2). My testing confirms this.

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the file system and the execution engine of the Infringing File System PLCs are adapted to load one or more recipe files into an executing industrial control program upon executing a load instruction in an industrial control program as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

313. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 5 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

f. Claim 6

314. Claim 6 reads as follows:

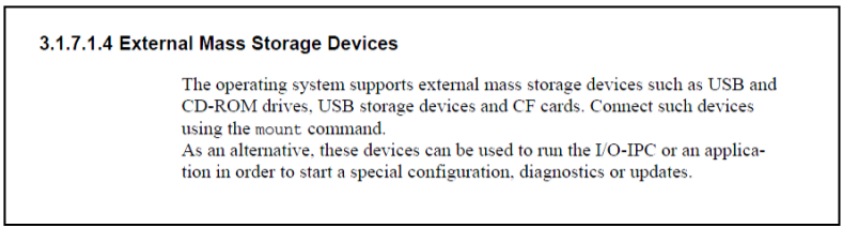
> 6. The system of claim 5, the recipe files being stored at a memory device separate from the program memory.

315. Claim 6 includes the element "the recipe files being stored at a memory device separate from the program memory." The Infringing File System PLCs can load recipe files that are stored at a memory device separate from the program memory.

316. Specifically, Claim 6 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. Recipe files can be stored at a separate device from the Infringing File System PLCs program memory, such as the controller's internal file

storage as well as USB memory sticks, Compact Flash, SD cards, external hard drives. This is confirmed by my testing.



3.1.7.1.4 External Mass Storage Devices

The operating system supports external mass storage devices such as USB and CD-ROM drives, USB storage devices and CF cards. Connect such devices using the `mount` command.
As an alternative, these devices can be used to run the I/O-IPC or an application in order to start a special configuration, diagnostics or updates.

(2004 System 758 Manual, p. 63); see also WAGO White Paper [ROCK0008605-ROCK0008607].

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs can load recipe files that are stored at a memory device separate from the program memory as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

317. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 6 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

g. Claim 7

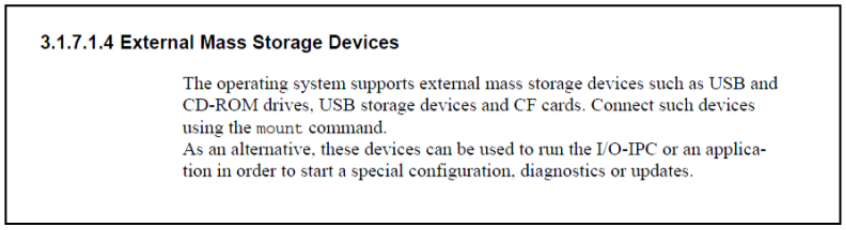
318. Claim 7 reads as follows:

> 7. The system of claim 6, the memory device being located at one of the industrial controller and a remote location from the industrial controller.

319. Claim 7 includes the element "the memory device being located at one of the industrial controller and a remote location from the industrial controller." The Infringing File System PLCs store recipe files in memory devices located on the Infringing File System PLC.

320. Specifically, Claim 7 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. Recipe files can be stored in a memory device internal to the Infringing File System PLC (separate from the program memory) in the PLC's internal storage, or in a USB memory stick, Compact Flash, SD cards, external hard drives.



3.1.7.1.4 External Mass Storage Devices

The operating system supports external mass storage devices such as USB and CD-ROM drives, USB storage devices and CF cards. Connect such devices using the `mount` command.
As an alternative, these devices can be used to run the I/O-IPC or an application in order to start a special configuration, diagnostics or updates.

(2004 System 758 Manual, p. 63); WAGO White Paper [ROCK0008605-ROCK0008607].

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs store recipe files in memory devices located on the Infringing File System PLC as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

321. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 7 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

h. Claim 10

322. Claim 10 reads as follows:

> 10. The system of claim 1, the file system and the execution engine being adapted to log measured data into a file upon executing an instruction in an industrial control program to record the measured data.

323. Claim 10 includes the element "the file system and the execution engine being adapted to log measured data into a file upon executing an instruction in an industrial control program to record the measured data." The Infringing File System PLCs have a file system and execution engine that can log measured data into a file upon executing an instruction in an industrial control program to record the measured data.

324. Specifically, Claim 10 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The WAGO manual entitled "Using the SysLibFile.lib and WAGOLibFTP.lib for file access" provides a coding example where measured data is stored in a file in an Infringing PLC. (See Application Note Version 1.0.0 pp. 6-7, 10 [ROCK0007865-66, 7869]).

b. The library, "Syslibfile.lib," is used to access the file system services. WAGO provides detailed instructions to use this library to save measured data to the file system. (See Application Note Version 1.0.0 pp. 6-7, 10 [ROCK0007865-66, 7869]).

c. WAGO's 30(b)(6) witness confirmed this element is present in the Infringing File System PLC's. (See Albers 30(b)(6) Dep. Tr. 84:22-85:4).

d. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs have a file system and execution engine that can log measured data into a file upon executing an instruction in an industrial control program to record the measured data as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

325. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 10 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

i. Claim 11

326. Claim 11 reads as follows:

> 11. The system of claim 10, the file system and the execution engine being adapted to retrieve measured data from a file upon executing an instruction in an industrial control program to load the measured data.

327. Claim 11 includes the element "the file system and the execution engine being adapted to retrieve measured data from a file upon executing an instruction in an industrial control program to load the measured data." The Infringing File System PLCs have a file system and execution engine that can retrieve measured data from a file upon executing an instruction in an industrial control program to load the measured data.

328. Specifically, Claim 11 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The program running on an Infringing PLC contains an instruction to access the file system to retrieve measured data. This is confirmed by my testing.

b. Moreover, the library, "Syslibfile.lib," provides the file service features to the Infringing File System PLC. WAGO provides detailed instructions to use this library to retrieve measured data from a file upon a load command. [ROCK0007871-72]

c. WAGO's 30(b)(6) witness confirmed this element is present in the Infringing File System PLC's. (See Albers 30(b)(6) Dep. Tr. 84:22-85:4, 91:1-8).

d. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs have a file system and execution engine that can retrieve measured data from a file upon executing an instruction in an industrial control program to load the measured data as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

329. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 11 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

j. Claim 12

330. Claim 12 reads as follows:

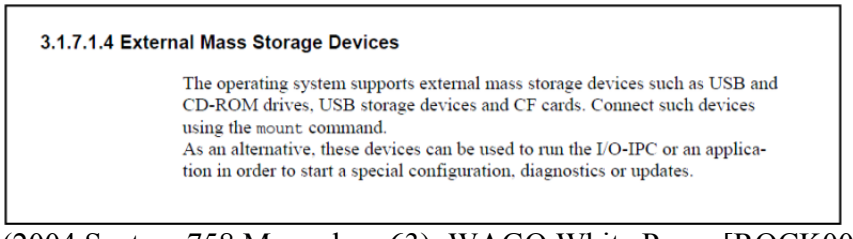
> 12. The system of claim 10, the measured data file being stored at a memory device separate from the program memory.

331. Claim 12 includes the element "the measured data file being stored at a memory device separate from the program memory." The measured data file that is logged by Infringing File System PLCs is stored at a memory device separate from the program memory.

332. Specifically, Claim 12 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The measured data that is stored by the Infringing File System PLCs is stored separate from the program memory. This is confirmed by my testing. (See also Albers 30(b)(6) Dep. Tr. 92:7-13).

b. The measured data that is stored by the Infringing File System PLCs can be stored in any location accessible by the PLC's file system, for example on a USB memory stick, Compact Flash card or SD card.



3.1.7.1.4 External Mass Storage Devices

The operating system supports external mass storage devices such as USB and CD-ROM drives, USB storage devices and CF cards. Connect such devices using the `mount` command.
As an alternative, these devices can be used to run the I/O-IPC or an application in order to start a special configuration, diagnostics or updates.

(2004 System 758 Manual, p. 63); WAGO White Paper [ROCK0008605-ROCK0008607].

> In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the measured data file that is logged by Infringing File System PLCs is stored at a memory device separate from the program memory as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

333. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 12 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

k. Claim 13

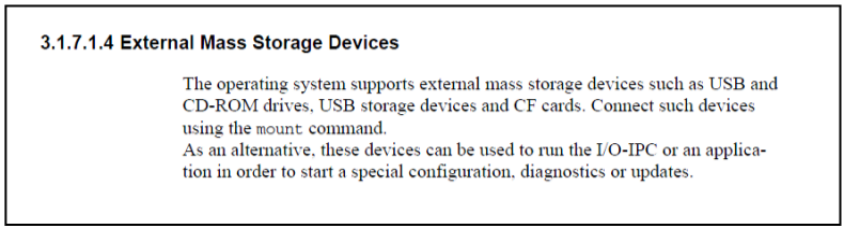
334. Claim 13 reads as follows:

> 13. The system of claim 12, the memory device being located at one of the industrial controller and a remote location from the industrial controller.

335. Claim 13 includes the element ", the memory device being located at one of the industrial controller and a remote location from the industrial controller." The measured data file that is logged by Infringing File System PLCs can be stored at a memory device located at least on the Infringing File System PLCs.

336. Specifically, Claim 13 is met literally, or in the alternative, under the doctrine of equivalents by WAGO-I/O System for at least the following reasons:

a. The measured data that is logged by the Infringing File System PLCs is stored on the PLC in the internal memory of the PLC, or on a USB memory stick, Compact Flash card or SD card.



3.1.7.1.4 External Mass Storage Devices

The operating system supports external mass storage devices such as USB and CD-ROM drives, USB storage devices and CF cards. Connect such devices using the `mount` command.
As an alternative, these devices can be used to run the I/O-IPC or an application in order to start a special configuration, diagnostics or updates.

(2004 System 758 Manual, p. 63); WAGO White Paper [ROCK0008605-ROCK0008607]

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the measured data file that is logged by Infringing File System PLCs can be stored at a memory device located at least on the Infringing File System PLCs as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

337. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 13 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

l. Claim 14

338. Claim 14 reads as follows:

> 14. The system of claim 1, the file system and the execution engine being adapted to log trend data into a file upon executing an instruction in an industrial control program to record the trend data.

339. Claim 14 includes the element "the file system and the execution engine being adapted to log trend data into a file upon executing an instruction in an industrial control program to record the trend data." The Infringing File System PLCs have a file system and

execution engine that logs trend data to a file upon executing an instruction in an Infringing File System PLC to record the trend data.

340. Specifically, Claim 14 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The Infringing PLC can execute a program directing trend data to be saved on the Infringing PLCs file system. (See Albers 30(b)(6) Dep. Tr. 68:1-4). My testing confirmed this.

Store trend data in the PLC	The trend handling in the PLC will be activated. The task TREND_TASK will be inserted automatically in the task configuration It will process an implicitly created ST-code for recording the trend data in a ring buffer and - if option History is activated in the trend element - for storing the values in a file system.

(CoDeSys Manual, p. 10-97)

7.1 Saving Trend Data

CoDeSys trend recording allows to save the written data in files on the WAGO-I/O-IPC. In the simplest case, the file can be created in the same file system ***"/data"*** directory in which the CoDeSys-program is installed on the IPC (see Fig. 7.1).

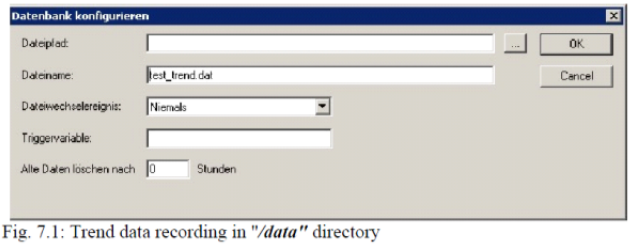


Fig. 7.1: Trend data recording in ***"/data"*** directory

(See WAGO-I/O-IPC Quickstart Manual at 30).

b. WAGO's 30(b)(6) witness confirmed this functionality. (See Albers 30(b)(6) Dep. Tr. 117:2-19; 758-870 Quickstart Manual Section 7 [ROCK0008347-ROCK0008348]).

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of

equivalents since the Infringing File System PLCs have a file system and execution engine that logs trend data to a file upon executing an instruction in an Infringing File System PLC to record the trend data as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

341. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 14 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

m. Claim 15

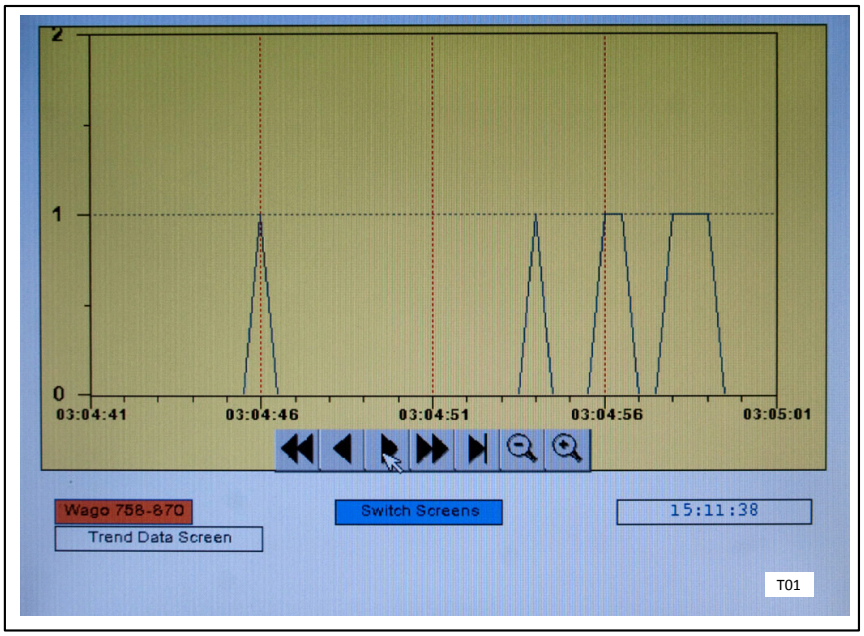
342. Claim 15 reads as follows:

> 15. The system of claim 14, the file system and the execution engine being adapted to retrieve trend data from a file upon executing an instruction in an industrial control program to load the trend data.

343. Claim 15 includes the element "the file system and the execution engine being adapted to log trend data into a file upon executing an instruction in an industrial control program to record the trend data." The Infringing File System PLCs have a file system and execution engine that retrieves trend data from a file upon executing an instruction in an Infringing File System PLC to load the trend data.

344. Specifically, Claim 15 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. A program executing on the Infringing File System PLCs can retrieve trend data from a file. My testing confirmed this.



(Trend data retrieved and displayed on 758-870 IPC)

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs have a file system and execution engine that retrieves trend data from a file upon executing an instruction in an Infringing File System PLC to load the trend data as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

345. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 15 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

n. Claim 16

346. Claim 16 reads as follows:

16. The system of claim 14, the trend data file being stored at a memory device separate from the program memory.

347. Claim 16 includes the element "the trend data file being stored at a memory device separate from the program memory." The Infringing File System PLC store trend data separate from the program memory.

348. Specifically, Claim 16 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The Infringing PLCs can execute a program directing trend data to be saved on the Infringing PLCs file system, separate from the program memory. This is confirmed by my testing. (See Albers 30(b)(6) Dep. Tr. 68:1-4)

Store trend data in the PLC	The trend handling in the PLC will be activated. The task TREND_TASK will be inserted automatically in the task configuration It will process an implicitly created ST-code for recording the trend data in a ring buffer and - if option History is activated in the trend element - for storing the values in a file system.

(CoDeSys Manual, p. 10-97)

7.1 Saving Trend Data

CoDeSys trend recording allows to save the written data in files on the WAGO-I/O-IPC. In the simplest case, the file can be created in the same file system *"/data"* directory in which the CoDeSys-program is installed on the IPC (see Fig. 7.1).

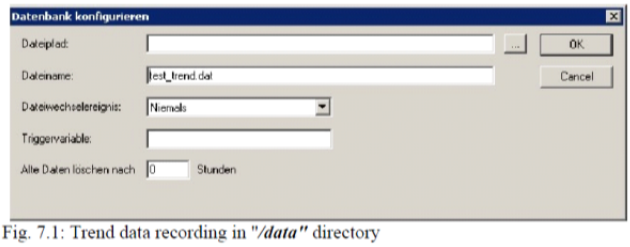


Fig. 7.1: Trend data recording in *"/data"* directory

(See WAGO-I/O-IPC Quickstart Manual at 30)

b. WAGO's 30(b)(6) witness confirmed this functionality. (See Albers 30(b)(6) Dep. Tr. 117:2-19; 758-870 Quickstart Manual Section 7 [ROCK0008347-ROCK0008348]).

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLC store trend data separate from the program memory as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

349. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 16 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

o. Claim 17

350. Claim 17 reads as follows:

> 17. The system of claim 16, the memory device being located at one of the industrial controller and a remote location from the industrial controller.

351. Claim 17 includes the element "the memory device being located at one of the industrial controller and a remote location from the industrial controller." The Infringing File System PLC stores trend data at least on a memory device located on the Infringing File System PLC.

352. Specifically, Claim 17 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The Infringing PLCs can execute a program directing trend data to be saved on the Infringing PLCs file system. See Albers 30(b)(6) Dep. Tr. 68:1-4;

Store trend data in the PLC	The trend handling in the PLC will be activated. The task TREND_TASK will be inserted automatically in the task configuration It will process an implicitly created ST-code for recording the trend data in a ring buffer and - if option History is activated in the trend element - for storing the values in a file system.

(CoDeSys Manual, p. 10-97)

7.1 Saving Trend Data

CoDeSys trend recording allows to save the written data in files on the WAGO-I/O-IPC. In the simplest case, the file can be created in the same file system "***/data***" directory in which the CoDeSys-program is installed on the IPC (see Fig. 7.1).

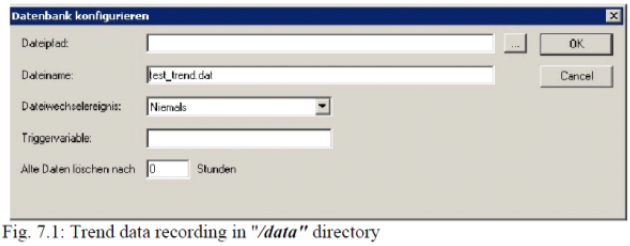


Fig. 7.1: Trend data recording in "***/data***" directory

(See WAGO-I/O-IPC Quickstart Manual at 30)

b. The Infringing Software Manual states that trend data is always stored in the directory of the CoDeSys program (CoDeSys Visualization Manual 2-44). This was confirmed in my testing. Alternately, the 758-870 IPC can store trend data in a user-defined file system location. (758-870 Quickstart Manual Section 7 [ROCK0008347-ROCK0008348])

c. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLC stores trend data at least

on a memory device located on the Infringing File System PLC as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

353. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claim 17 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

p. Claim 20

354. Claim 20 reads as follows:

> 20. The system of claim 1, the industrial control program being a ladder logic program.

355. Claim 20 includes the element "the industrial control program being a ladder logic program." The industrial control program stored on the Infringing File System PLCs contains ladder logic instructions.

356. Claim 20 is met literally, or in the alternative, under the doctrine of equivalents by the Infringing File System PLCs for at least the following reasons:

a. WAGO's 30(b)(6) witness agreed with the assertion that a WAGO customer could write a program "containing Ladder Logic instructions to employ the file system on the WAGO industrial controller to log data to a file." (See Albers 30(b)(6) Dep. Tr. 84:22-85:11).

b. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the industrial control program stored on the Infringing File System PLCs contains ladder logic instructions as set forth herein,

and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

357. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 20 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

q. Claim 21

358. Claim 21 reads as follows:

> 21. A method for providing, an industrial controller with the functionality associated with utilizing a file system residing in the industrial controller, the method comprising:
>
> [1] developing a file system and loading the file system on an industrial controller, the file system having a plurality of file system services; and
>
> [2] developing an execution engine that interprets instructions of an industrial control program that utilizes at least one of the plurality of file system services.

359. Claim 21 of the '813 patent is directed to "A method for providing, an industrial controller with the functionality associated with utilizing a file system residing in the industrial controller, the method comprising". I have been informed that this portion of the claim is called the preamble and does not give life and meaning to the claims, and therefore does not serve to limit the claims. Accordingly, I will not address the preamble in detail in this report because it is not a limitation in the claim. However, it is my opinion that WAGO does practice a method for providing an industrial controller with the functionality associated with utilizing a file system residing in the industrial controller.

(i) Element [1] of Claim 21

360. Element [1] of Claim 21 includes "developing a file system and loading the file system on an industrial controller, the file system having a plurality of file system services." The Infringing File System PLCs have a file system with a plurality of file system services.

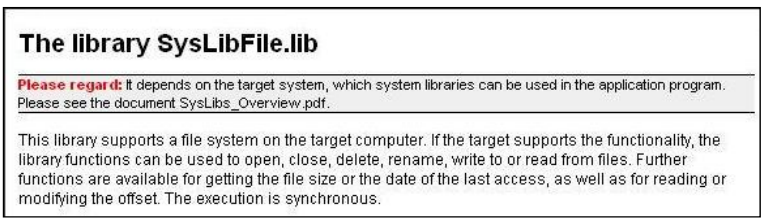
361. Specifically, Claim 21 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The Infringing File System PLCs comprise those Programmable Field Bus Controllers (750 controllers), and IPC products (758 controllers) that contain file systems residing in the program memory of those controllers. (See www.wago.us/produts/37712.htm; Albers 30(b)(6) Dep. Tr. 71:4-15; 71:25-72:5).

b. The first WAGO-I/O System controller that had an operating system was the 750-841 controller. (See Albers 30(b)(6) Dep. Tr. 73:12-16).

c. I am informed that the WAGO 750-841 was first released for sale sometime in 2003. (See Albers 30(b)(6) Dep. Tr. 73:22-25).

d. The file system on an Infringing File System PLC has a plurality of file system services that can be accessed via various SysLibFile.lib functions. See "Using the SysLibFile.lib and the WagoLibFtp.lib for file access" Application Note Version 1.0.0 pp. 6-7, 10 [ROCK0007865-66, 7869].



The library SysLibFile.lib

Please regard: It depends on the target system, which system libraries can be used in the application program. Please see the document SysLibs_Overview.pdf.

This library supports a file system on the target computer. If the target supports the functionality, the library functions can be used to open, close, delete, rename, write to or read from files. Further functions are available for getting the file size or the date of the last access, as well as for reading or modifying the offset. The execution is synchronous.

(CoDeSys v.2.3 program help; see also other file system services are included in other libraries. (See Controller Manuals and CoDeSys 2.3 Manual)

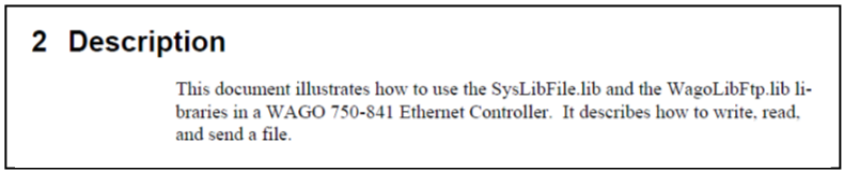
In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs have a file system with a plurality of file system services as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

(ii) Element [2] of Claim 21

362. Element [2] of Claim 21 includes "developing an execution engine that interprets instructions of an industrial control program that utilizes at least one of the plurality of file system services." The Infringing File System PLCs have an execution engine that interprets instructions of an industrial control program that utilizes at least one of the plurality of file system services.

363. Specifically, Claim 21 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. My testing showed that the Infringing File System PLCs use portions of the Infringing Software as an execution engine that interprets instructions in an industrial control program to use file system services in the controller.



2 Description

This document illustrates how to use the SysLibFile.lib and the WagoLibFtp.lib libraries in a WAGO 750-841 Ethernet Controller. It describes how to write, read, and send a file.

(Wago Application note A114100 Page 6)

b. I am told that the execution engine runtime system is installed on the Infringing File System PLCs prior to the download of the control program. (See Albers 30(b)(6) Dep. Tr. at 86:5-15; see also CoDeSys Brochure WCP10006).

c. The execution engine runtime system resides in the program memory of the controllers. (See Albers 30(b)(6) Dep. Tr. 78:23-79:80-12, 142:18-143:3).

d. The runtime system is adapted to interpret instructions from an industrial control program, the industrial control program including at least one instruction utilizing one or more of the plurality of file system services. (See Albers 30(b)(6) Dep. Tr. 78:8-12, 80:12; 86:5-15; 142:7-12).

e. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing File System PLCs have an execution engine that interprets instructions of an industrial control program that

utilizes at least one of the plurality of file system services as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

364. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing Claim 21 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

365. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

366. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, these components have no substantial, non-infringing uses, and the components constitute a material part of Rockwell's invention. WAGO is aware of the '813 patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '813 patent.

367. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 21 of the '813 patent.

368. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 21 of the '813 patent.

r. Claim 22

369. Claim 22 reads as follows:

> 22. The method of claim 21, further comprising developing an industrial control program including at least one instruction that utilizes one or more file system services and downloading the industrial control program to the industrial controller.

370. Claim 22 includes the element "developing an industrial control program including at least one instruction that utilizes one or more file system services and downloading the industrial control program to the industrial controller." The Infringing Software is used to develop industrial control programs containing one ore more instructions that utilize the file system services of the Infringing File System PLCs, which are downloaded onto the Infringing File System PLCs.

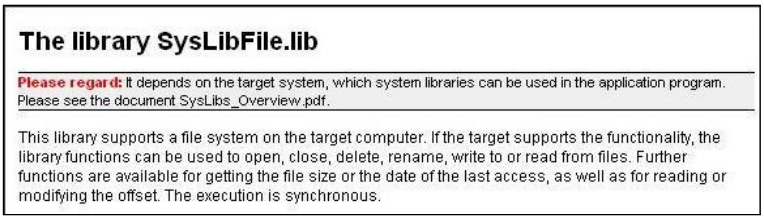
371. Specifically, Claim 22 is met literally, or in the alternative, under the doctrine of equivalents by the WAGO-I/O System for at least the following reasons:

a. The Infringing File System PLCs are programmed with the Infringing Software.

b. A program can be downloaded to the Infringing File System PLCs.

c. The Infringing Software contains an editor to write industrial control programs.

d. The file system on an Infringing File System PLC has a plurality of file system services, which can be utilized by the SysLibFile.lib function. See "Using the SysLibFile.lib and the WagoLibFtp.lib for file access" Application Note Version 1.0.0 pp. 6-7, 10 [ROCK0007865-66, 7869].



(CoDeSys v.2.3 program help)

In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the Infringing Software is used to develop industrial control programs containing one or more instructions that utilize the file system services of the Infringing File System PLCs, which are downloaded onto the Infringing File System PLCs as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

372. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has literally infringed and is infringing Claim 22 of the '813 patent literally, or in the alternative, under the doctrine of equivalents.

373. Given the expected uses of the WAGO-I/O System products and the instructions and guidance provided by WAGO, it is my opinion that WAGO customers have used and are using those products in the manner described above. It is also my opinion that WAGO knew or should have known that its customers would do so.

374. It also is my opinion that the components described above are sold by WAGO for use in an infringing method, these components have no substantial, non-infringing uses, and the components constitute a material part of Rockwell's invention. WAGO is aware of the '813

patent and knows that the methods for which these components have no other substantial use may be covered by a claim of the patent or may satisfy a claim under the doctrine of equivalents, and that use of the components directly infringes the '813 patent.

375. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO has actively induced infringement and is actively inducing infringement of Claim 22 of the '813 patent.

376. Based on the foregoing, the attached Exhibits, and based on other information disclosed in this report, it is my opinion that WAGO contributes to their customers' direct infringement of Claim 22 of the '813 patent.

4. WAGO's Infringement of U.S. Patent No. 7,065,415

377. I understand that Plaintiffs are asserting claims 1-5, and 8 of the '415 patent. It is my opinion that WAGO has been and is directly infringing claims 1-5, and 8 either literally or under the doctrine of equivalents based on the admissions of WAGO's 30(b)(6) witness.

378. I do not believe any of the claim terms in the '415 patent require explanation or a Court's interpretation, other than the term "recipe file," as the remainder of the terms are clear and easily understood not only by a PHOSITA, but simply by laypeople as well. It is my opinion that the terms of the '415 patent are understood in light of the patent's specification and the claims themselves, such that no construction should be adopted that alters the plain meaning of these words.

379. If the Court construes the term "recipe file" as used in the '813 patent, it should mean "a profile and associated parameters regarding a particular implementation or process."

This definition is supported by my experience in the field and the intrinsic evidence in '813 patent and its prosecution history.

380. While the term "control operation" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "control operation," it is my opinion that the claim elements are satisfied for infringement.

381. While the term "editor for developing ladder logic programs" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "editor for developing ladder logic programs," it is my opinion that the claim elements are satisfied for infringement.

382. While the term "execution engine" as used in the '813 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '813 patent and its prosecution history. Defendants' definition is not supported by the

intrinsic evidence of the '813 patent. Defendants attempt to limit execution engine to an "interpreter" which is in my opinion too narrow of a construction based on the way the term is used in the '813 patent, and also based on my experience with compiled and interpreted computer processing.

383. While the term "file system" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "file system," it is my opinion that the claim elements are satisfied for infringement.

384. While the term "file system services" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "file system services," it is my opinion that the claim elements are satisfied for infringement.

385. While the term "header instructions" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my

experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "header instructions," it is my opinion that the claim elements are satisfied for infringement.

386. While the term "industrial control program" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "industrial control program," it is my opinion that the claim elements are satisfied for infringement.

387. While the term "industrial controller" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. Moreover, Plaintiffs' position supports the notion that "industrial controller" in the context of the patent means a PLC. However, taking either party's construction of the term "industrial controller," it is my opinion that the claim elements are satisfied for infringement.

388. While the term "ladder logic" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs'

construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "ladder logic," it is my opinion that the claim elements are satisfied for infringement.

389. While the term "load instruction" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field. and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "load instruction," it is my opinion that the claim elements are satisfied for infringement.

390. While the term "program memory" as used in the '415 patent has plain and ordinary meaning and should be construed in accordance with that meaning, if the Court were to adopt one of the party's proposed claim constructions on this term, it is my opinion that the Plaintiffs' construction is more appropriate than Defendants' construction based on my experience in the field, and because Plaintiffs' position is supported by the intrinsic evidence in the '415 patent and its prosecution history. However, taking either party's construction of the term "program memory," it is my opinion that the claim elements are satisfied for infringement.

391. The asserted claims of the '415 patent are directed to "An editor for developing ladder logic programs that control operation of an industrial controller system, the editor comprising". It is my opinion that the WAGO-I/O System provides an editor for developing ladder logic programs that control operation of an industrial controller system based on WAGO's 30(b)(6) deponent, Dr. Thomas Albers, admission of infringement of each and every element of

the asserted claims of the '415 patent. (See Albers 30(b)(6) Dep. 84:22-85:4; 85:6-11; 77:21-80:12; 85:13-86:15; 84:22-85:4, 91:1-8; 68:1-4; 80:25-81:3; 78:8-12:80-12; 86:5-15; 142:7-12).

392. In the event that claim interpretation by the Court should result in no literal infringement, it is my view that any reasonable interpretation of this claim element would still result in infringement under the doctrine of equivalents since the WAGO-I/O System provides an editor for developing ladder logic programs that control operation of an industrial controller system as set forth herein, and would thus perform substantially the same function in substantially the same way to yield substantially the same result.

393. Based on the foregoing, and based on other information disclosed in this report, it is my opinion that WAGO has infringed and is infringing claims 1-5 and 8 of the '415 patent literally, or in the alternative, under the doctrine of equivalents.

VII. AUTHOR'S QUALIFICATIONS, COMPENSATION, AND OTHER BACKGROUND INFORMATION

394. A description of my work and educational experience is contained in my *curriculum vitae*, which is attached as Exhibit A.

395. I am an engineering consultant now working for my own company, Artzat Consulting, LLC. Between 1997 and 2009, while working part-time as an engineering consultant, my primary employment was as an engineer and executive with other companies. Although now consulting full time, I still maintain an adjunct relationship with a control systems integration company as detailed below.

396. My consulting work generally involves technical and commercial matters related to intellectual property, accidents and damages, and design & operations. The practice areas in which I work include industrial equipment, electrical and electronic engineering, computers & computerized devices, industrial automation & control, and construction.

397. I received the a Bachelor of Science degree in Engineering from the University of New Orleans in 1975, and a Master of Science degree in Engineering from the University of New Orleans in 1979.

398. I am a Registered Professional Control Systems Engineer in Louisiana (1993). I am also a registered Professional Electrical Engineer in Louisiana (1980), California (1992), Alaska (1992), and Alabama (1997).

399. I am a Certified Member (#774) of the National Academy of Forensic Engineers (NAFE), and a Licensed Member of the National Society of Professional Engineers (NSPE). I am a Senior Member of The International Society of Automation (ISA), and of the Institute of Industrial Engineers (IIE). I am a Professional Member of the American Society of Safety Engineers (ASSE). I am a Member of the Institute of Electrical and Electronic Engineers (IEEE) and of the Louisiana Engineering Society.

400. I was certified as a Systems Integrator by the Institute of Industrial Engineers (1991), and was certified in Computer Forensics at Oregon State University (2004). I am a Licensed Master Electrician in the State of Texas (2009), and I am (or have been) a Qualifying Party for Contractor in Louisiana, Texas, Tennessee, and Mississippi.

401. I am a co-inventor of U.S. Patent No. 5,103,395, entitled "System for Remote Positioning of a Radioactive Source into a Patient Including Means for Protection Against Improper Patient Exposure to Radiation," which issued on April 7, 1992.

402. Since April 2009, I have maintained an adjunct relationship with VersaTech Automation Services as a Senior Consultant. I serve in an advisory capacity for design, manufacture, and service of industrial control and automation systems. The company designs and

integrates systems for a variety of mechanical and process applications, with operations in Houston, New Orleans, and Dubai.

403. During 2008 and 2009, I was the Director of Stewart Resource Center. This is the in-house Architecture and Engineering entity for projects at the 300+ commercial properties owned and operated by Stewart Enterprises, Inc.

404. From 1988 to 2008, I worked in senior technical and management positions at TEST Automation & Controls, Inc. ("TEST"). TEST was a publicly-held manufacturing firm dealing with industrial automation and controls. TEST provided design, manufacturing, and construction for electrical and instrumentation projects worldwide. The related systems used electrical, electronic, pneumatic, and hydraulic technologies. I joined TEST via its acquisition of Dataran in 1988, and retired as its Vice President in 2008. TEST was subsequently acquired by Cameron Flow Control in 2009.

405. From 1980 to 1987, I was a Principal Engineer at Dataran Corporation. This professional engineering firm designed and managed the installation of industrial power, control, and automation systems. Dataran also developed and manufactured high performance PC "clone" computers specially configured for industrial and CAD applications. The proprietary industrial control technology evolved into TEST's SCADAWARE® telemetry product line when Dataran was acquired by TEST in January of 1988.

406. From 1977 to 1980, I was an Electrical and Electronic Engineer for Shell Offshore. This was a specialist position for technologies involving power, automation, safety and control systems, and communications. I also conducted training sessions related to controls, instrumentation, and computer programming while at Shell, and also after starting Dataran.

407. From 1977 to 1981, I was a full-time and part-time instructor of Engineering in both the Electrical and Mechanical departments at the University of New Orleans School Of Engineering. I taught courses in basic electricity, electronics, mechanics, and fluids.

408. From 1976 to 1977, I was a Design & Construction Engineer for SECO Industries. That position involved the design and project management for power distribution, process controls, and safety systems, with emphasis on industrial and offshore oilfield environments.

409. From 1975 to 1976, I was an Industrial Equipment Engineer for Boeing Services. This position involved design and specification of maintenance-related equipment, both mechanical and electrical, for NASA's Michoud Assembly Plant in New Orleans.

410. From 1973 to 1974, I was a Medical Equipment Service Technician for Coulter Electronics. In that position, I installed and maintained automated particle counting and blood analysis equipment for industrial and medical laboratories.

411. A list of publications authored or co-authored by me in the previous ten years and presentations that I have given is contained in my *curriculum vitae*, attached as Exhibit A.

412. A list of all other cases in which I have testified as an expert at trial or by deposition in the last four years (2008-2/24/2012) is attached as Exhibit A.

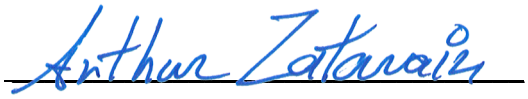
413. My compensation for the work I perform on this lawsuit is $320 per hour. My compensation does not depend on the outcome of this lawsuit.

414. I have not yet prepared exhibits to be used at trial as a summary of, or support for, the opinions expressed in this report, but I expect to do so in accordance with the Court's scheduling orders.

VIII. SUPPLEMENTATION

I reserve the right to revise, supplement, or amend my opinions contained in this report in light of any additional information that I might receive after the date of this report including, but not limited to, reports submitted by other experts in this lawsuit.

Dated: February 24, 2012



Arthur Zatarain, PE

CERTIFICATE OF SERVICE

I hereby certify that on this 24th day of February, 2012, pursuant to a written agreement between the parties' counsel governing service in this action, a copy of this document is being served by electronic mail on Defendants' counsel listed below at the email addresses listed below:

Robert Cook, Esq.
Whitham, Curtis, Christofferson & Cook, P.C.
11491 Sunset Hills Road, Suite 340
Reston, Virginia 20190
Tel.: (703) 787-9400, ext. 122
Fax: (703) 787-7557
Email: bob@wcc-ip.com

John C. Scheller, Esq. (SBN 1031247)
Michael Best & Friedrich LLP
One South Pinckney Street – Suite 700
Madison, WI 53703
Tel.: (608) 257-3501
Fax: (608) 283-2275
Email: jcscheller@michaelbest.com

/s/ Brian Pollack
Brian Pollack

www.artzat.com

Artzat Consulting, LLC
401 Arlington Dr.
Metairie, LA. 70001 USA
Engineering and Operations Consulting

(504) 837-3090
Central Time
arthur@artzat.com

Arthur Zatarain, PE

Equipment, Controls, Instrumentation,
and Electrical / Electronic Systems

October 2011

Exhibit A

CURRENT AFFILIATIONS

Artzat Consulting, LLC - 1997 to Present (concurrent with fulltime positions prior to Sep 2009)
Engineering Consultant
Consulting for technical and commercial matters related to intellectual property, accidents and damages, and design & operations. Practice areas include industrial equipment, electrical and electronic engineering, computers & computerized devices, industrial automation & control, and construction. Fulltime consulting since Sept 2009.

VersaTech Automation Services – April 2009 to Present
Senior Consultant
Serve in an adjunct advisory capacity for design, manufacture, and service of industrial control and automation systems for a variety of mechanical and process applications, with operations in Houston, New Orleans, and Dubai.

PRIOR CAREER EXPERIENCE

Stewart Enterprises Inc. – July 2008 to Sept 2009
Director of Stewart Resource Center
Stewart Resource Center (SRC) is the in-house Architecture and Engineering entity that designs and manages the construction of 300+ commercial properties owned and operated by Stewart Enterprises, Inc. throughout North America.

TEST Automation & Controls, Inc. (TEST Inc.) – 1988-2008
1997-2008: VP- Operations
1995-1997: VP - Engineering
1988-1995: Engineering Manager
TEST is a publicly held industrial manufacturing firm dealing with automation and controls, providing design, manufacturing, and construction for projects worldwide. The automation systems used electrical, electronic, pneumatic, and hydraulic technologies. Joined TEST via acquisition of Dataran in 1988. Retired from TEST in July 2008. The company, as a division of Natco (NYSE:NTG), was absorbed into Cameron International (NYSE:CAM) in Nov 2009.

Dataran Corporation - 1980-1987 (acquired by TEST in '88)
Principal Engineer
A professional engineering consulting firm that designed and managed the installation of industrial power, control, and automation systems. Also developed and manufactured high performance PC "clone" computers specially configured for industrial and CAD applications. The proprietary technology evolved into TEST's SCADAWARE® telemetry product line when acquired by TEST in January of 1988.

Shell Offshore - 1977-1980
Electrical & Electronic Engineer
Technical Specialist for major oil producer. Duties included design for power, automation, safety and control systems, and communications. Also conducted training sessions related to controls, instrumentation, and computer programming.

University of New Orleans School of Engineering - (full and part time 1977-81)
Served as an instructor of Engineering at UNO in both the Electrical and Mechanical departments while working at Shell and Dataran. Courses included basic electricity, mechanics, fluids, and electronics.

SECO Industries - 1976-1977
Design & Construction Engineer
Design and project engineer for power distribution, controls, and safety systems, focusing on industrial and offshore oilfield environments.

Boeing Services - 1975-1976
Industrial Equipment Engineer
Design and specification of maintenance related equipment, both mechanical and electrical for NASA's Michoud Assembly Plant in New Orleans.

Coulter Electronics - 1973-74
Medical Equipment Service Technician
Installed and maintained particle counting and blood analysis equipment for industrial and medical laboratories.

ACADEMIC & PROFESSIONAL ACHIEVEMENTS

- University of New Orleans, B.S. Engineering 1975, M.S. Engineering 1979
- Registered Professional Electrical Engineer: LA (1980), CA (1992), AK (1992), AL (1997)
- Registered Professional Control Systems Engineer: Louisiana (1993)
- Certified Member: National Academy of Forensic Engineers (NAFE)
- Licensed Member: National Society of Professional Engineers (NSPE)
- Senior Member: The International Society of Automation (ISA)
- Senior Member: Institute of Industrial Engineers (IIE)
- Professional Member: American Society of Safety Engineers (ASSE)
- Member: Institute of Electrical and Electronic Engineers (IEEE)
- Member: Louisiana Engineering Society
- Certified Systems Integrator - Institute of Industrial Engineers (1991)
- Certified in Computer Forensics – Oregon State University (2004)
- Licensed Master Electrician, Texas. (2009)
- Qualifying Party for Contractor in Louisiana, Texas, Tennessee, Mississippi
- U.S. Patent No. 5,103,395 for Automated Nuclear Medical Device, (1992)

PUBLICATIONS AND PRESENTATIONS

- Don't Gamble with your Safety Instrumented System, Plant Services Magazine, July 2010
- Be Smart about Intellectual Property, Plant Services Magazine, October 2009
- Testing Redundant Systems, Plant Services Magazine, March 2006
- Control Systems Reliability, Plant Services Magazine, March 2005
- Safety and Procurement Management, US Bus Review, Mar 2004 (w/Bill Smitka)
- PC Based SCADA Systems - Trinidad and Tobago - 1995
- Smart RTU & SCADA for Offshore - Abu Dhabi, UAE - 1993
- Shell Oil SCADA Instrumentation, Annually 1977 thru 1993
- Packet Radio for SCADA Applications, ENTELEC Conference, Dallas, 1992
- PC Based Data Acquisition, Am. Soc. of Quality Engrs., Atlantic City, 1986
- Personal Computers for Technical Users, ISA Philadelphia, Oct 1985
- Offshore Safety Systems, ISA '83, Houston, Texas (w/ Bob Bailliet).
- Economics of Wind Generated Power, ASME 1978, Houston, TX. (w/ Bill Janna)

-end-

Deposition and Trial Testimony of Arthur Zatarain, PE

2008 to 2/24/2011

1. Deposition – 12/21/2011 Individuals, Transocean, Halliburton, Cameron, Swaco, Weatherford, Anadarko, Moex and US Government v **BP**, USDC Eastern District of Louisiana, MDL 2179

2. Deposition - 12/14/2011 - **Micro Motion, Inc.** v. Krohne, Inc., Krohne Messtechnik GmbH & Co. KG, USDC Massachusetts, Civil Action No. 09-CV-10319-NG

3. Court Testimony – 6/8/2011 **Key Energy** v CC Forbes and Petron Industries, USDC Eastern Texas, Case No. 2:08-CV-346-DF-CE.

4. Deposition – 4/21/2011 **Key Energy** v CC Forbes and Petron Industries, USDC Eastern Texas, Case No.. 2:08-CV-346-DF-CE.

5. Deposition: 4/4/2011 Matt Koessel v Hyundai Motor Manufacturing…**Meta Control**, Circuit Court of Montgomery AL, Case CV-08-865

6. Deposition: 2/9/2011 LA County Sanitation District No 2 v **Siemens Power Generation**, Superior court of California, Orange County, Case No. 30-2009-00330264.

7. Deposition: 1/7/2011 TI Pneumotive v Ecological Tanks and **Yasunaga Corp**, Parish of Ouachita, Louisiana, Case No. 04-1128.

8. Deposition (video): 4/8/2010 Gilbarco v **Franklin**, USDC of North Carolina, Greensboro Div., Civil action 1:06-000585-JAB-PTS

9. Deposition (video): 3/31/2010 Gilbarco v **Franklin**, USDC of North Carolina, Greensboro Div., Civil action 1:06-000585-JAB-PTS

10. Deposition: 3/26/2011 MicroMotion v Krohne, USDC of Massachusetts, Civil Action No. 09-CV10319-NG.

11. Deposition: 3/10/2009 Clinton C Holman Jr v **Blout International Inc** et al, Circuit court of Jefferson County, Alabama, 01-CV-2006-001873.00

12. Deposition: 6/17/2008 **Oceaneering Int'l** v GRI Simulations, USDC Western Dist LA, Civil Action No CV05-0258

EXHIBIT B

List of Documents Reviewed

Document	Bates No.
WAGO I/O SYSTEM - Full Line Catalog 2010/2011	
WAGO-I/O-PRO CAA Compatibility Chart	ROCK0007738-ROCK0007744
CoDeSys 2.3 Manual	ROCK0007278-ROCK0007737
CoDeSys V3.x User Documentation	ROCK0007745-7762
750-841 Manual Section 3.1.7	ROCK0008463
758-870/000-110 Manual Section 11-12	ROCK0008687-ROCK0008720
WAGO Speedway 767-2301 and 767-2501 Manuals	
C-Code in Codesys	
Creating and Linking External C Library Functions - Document V1.0	
Ds Response to 2d Set Ints (with exhibits)	
Ds Response to Ps 1st Set of Ints (2011-10-07)	
Ds Resonse to 1st Set of Requests for Admission (2012-02-02)	
Ds Response to Ps 3d Set of Ints (20111-12-21)	
Ds Rule 26(a)(1) Disclosures	
Supp to Ds Non-Infringement Contentions (with exhibits) (2011-11-22)	
750-841 Powerpoint	ROCK0006768-ROCK0006823
Codesys Powerpoint	ROCK0006824-ROCK0006861
CoDeSys Brochure	WCP 10001-20
	WCP10021-24
	WCP22688-WCP22705
	WCP22837-WCP22855
750-880 Datasheet	ROCK0006862-ROCK0006863

750-880 Manual	ROCK0006864-ROCK0007277
Codesys 2.3 Manual	ROCK0007278-ROCK0007737
Codesys Compatability	ROCK0007738-ROCK0007744
8880 Powerpoint	ROCK0007785-ROCK0007784
Ethernet Controller Brochure	ROCK0007785-ROCK0007812
	ROCK0007813-ROCK0007826
IPC Brochure PPT	ROCK0007851-ROCK0007859
SysFileLib	ROCK0007860-ROCK0007873
758-870xxx Manual	ROCK0007894-ROCK0008127
758-874 Manual	ROCK0008128-ROCK0008317
758-870 Quickstart (trend)	ROCK0008318-ROCK0008349
Pro32 Compatability	ROCK0008350-ROCK0008350
750-841 Manual	ROCK0008375-ROCK0008604
Whitepaper	ROCK0008605-ROCK0008607
Deposition Transcripts fo Rinn, Albers, Decramer, Artmann	
Patents and related prosecution histories of the ‘232, ‘974, ‘813, ‘415 patents	

Wago I/O System Test Notes

Jan-Feb 2012

1. This report was prepared by Arthur Zatarain, Artzat Consulting, Metairie LA. Contact info 504-837-3090, artzat.com.

2. The Artzat project number was 2012-324. Testing was conducted at the office of Artzat Consulting in Metairie, LA. All testing and documentation was prepared by Arthur Zatarain.

3. This report provides test notes and related information on testing conducted in Feb 2012 on several WAGO industrial control devices. The testing was done in relation to litigation regarding infringement of several patents owned by Rockwell.

4. The following tested WAGO devices relevant to the related litigation were as follows:

   4.1. I/O-IPC-G2

      4.1.1. Unit marked with 758-870/0000-0110, s/n 139668009, ID 1028-5489, HW 10 Rev 00, 27.05.09

      4.1.2. Unit has CoDeSys holographic sticker on the bottom, s/n 100293154

      4.1.3. Fitted with 750-600 8 point digital I/O slice addressed as IX0.0-0.8 and QX0.0-0.8.

      4.1.4. Network connection to port X8 at 192.168.1.17.

      4.1.5. Alt-F2 gets to command console, log in to root with password=wago.

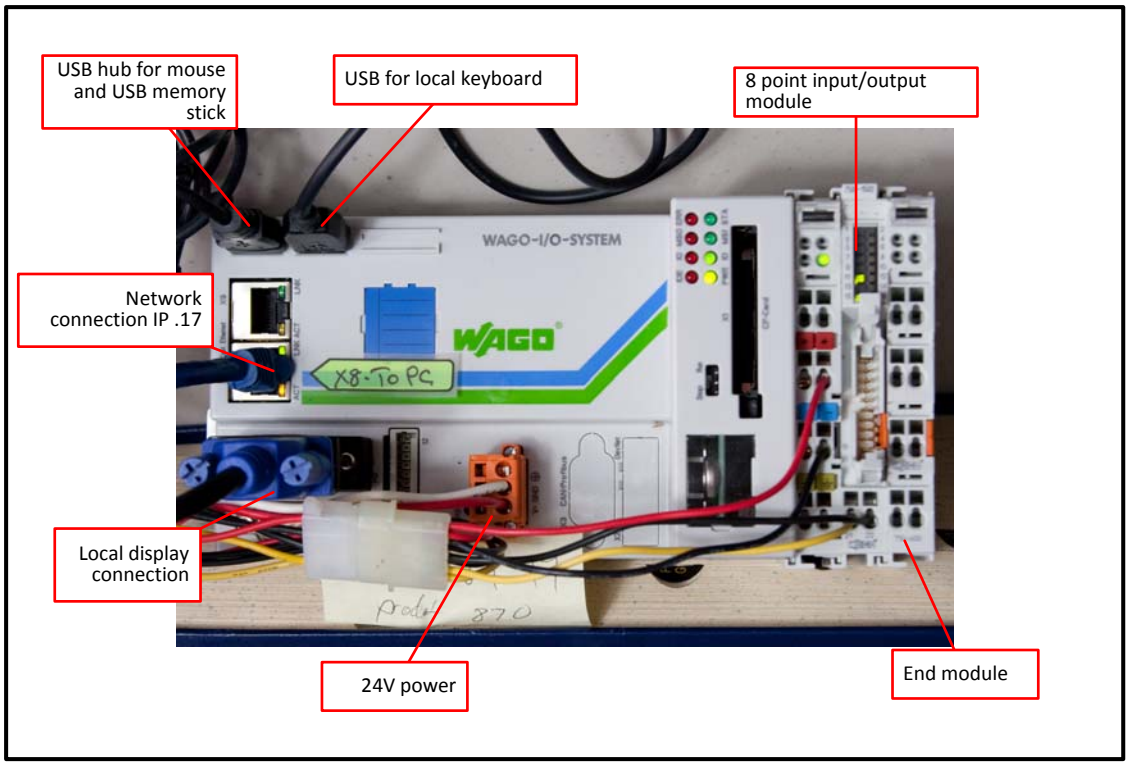
      4.1.6. Alt-F1 gets to VGA screen from Codesys. Alt-F3 gets to config screen.

      4.1.7. CoDeSys runtime plclinux_rt was located in dir /usr/sbin. The file is approx. 1.7Mb, dated 1/25/2011.

      4.1.8. Codesys working dir seems to be /home/codesys.

      4.1.9. External USB drive is mounted by Linux at /media/sda1.

      4.1.10. Attached I/O was an 8 input / 8 output module.



WAGO 758-870 IPC

4.2. WAGO-I/O-IPC

4.2.1. Unit marked with 758-870, s/n 151537001, ID 27173, Rev 14, 25/11/10

4.2.2. This unit does not have a CoDeSys holographic sticker on the bottom side.

4.2.3. IP address set to 192.168.1.17 on X9 (not X8 as on the other 870).

4.2.4. After contacting Wago support, was able to log in to Linux console root, case sensitive password = Ko2003wa.

4.2.5. Could not get CoDeSys to interface with this unit (CoDeSys runtime likely not installed). No related files noted in the directory system.

4.3. Wago 750-880 Controller

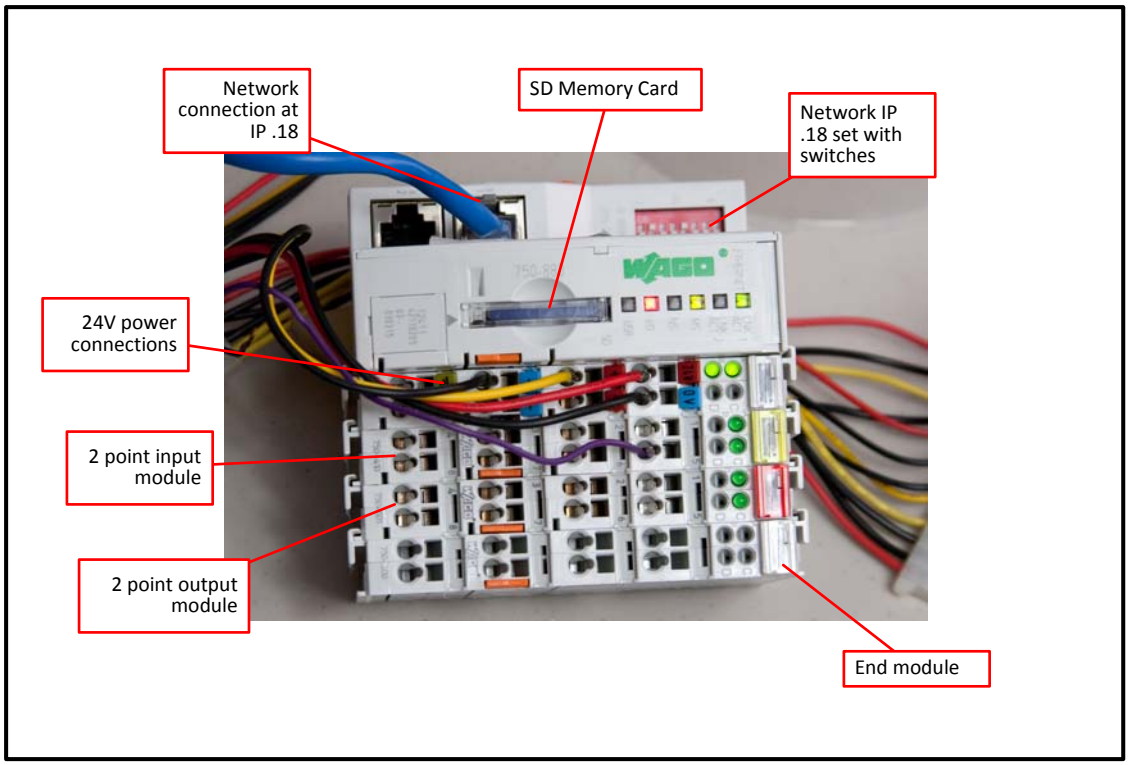
4.3.1. Unit marked with firmware 01.01.07 (02).

4.3.2. IP address set to 192.168.1.18 using dipswitch.

4.3.3. Login to user name Admin, password = wago.

4.3.4. 512MB SD memory card provided by author

4.3.5. Attached I/O was a 2 point input module and a 2 point output module.



WAGO 750-880 Controller

4.4. Wago Software

4.4.1. Wago-I/O-Pro CAA

4.4.2. CoDeSys V 2.3

5. Controller Test Configuration

5.1. The testing environment for the Wago I/O System devices included the following

5.1.1. A local area network using 100mb Ethernet with a 5 point hub.

5.1.2. 24 V power supply

5.1.3. Personal computers (details provided below)

5.1.4. Basic test equipment such as volt meter, small tools, jumper wires.

5.1.5. Computer display monitors, keyboards, and mice (detailed below where appropriate)

5.1.6. Wago 758-870 and 750-880 controllers on the same network with the personal computer.

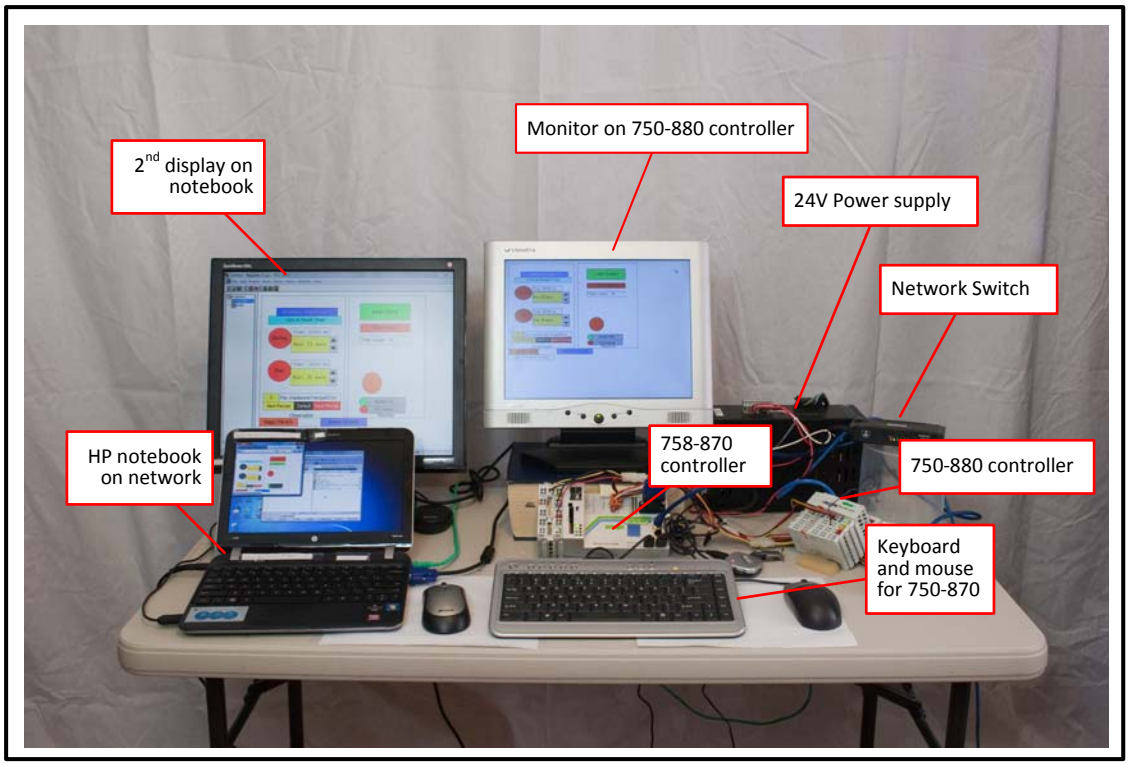


Photo of test configuration

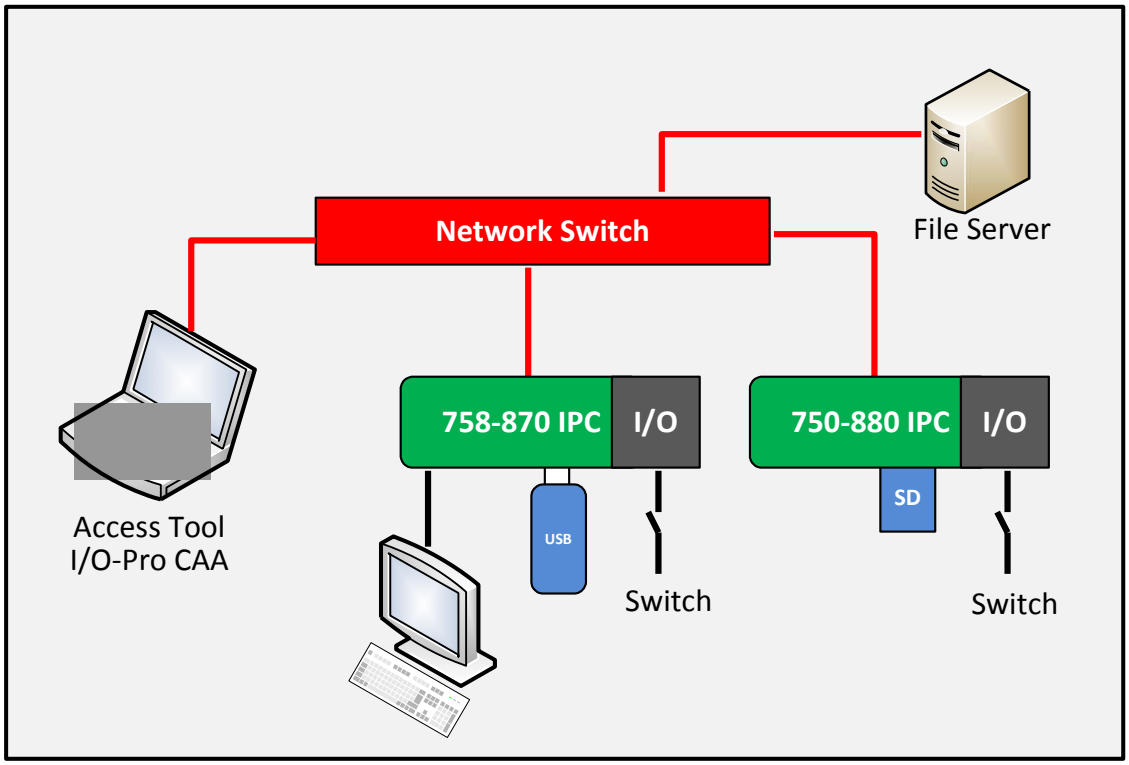


Diagram of test configuration

5.2. Personal Computers

5.2.1. Initial testing was done with a Dell notebook computer provided by the author. Final testing as documented herein was done with a dedicated HP notebook provided by Plaintiff counsel. The relevant software and configuration of both computers was similar (or identical) as noted below.

5.2.2. The test computers had a Windows 7 operating system, Total Commander file manager, and Ubuntu linux operating in a Oracle Virtual Box environment.

5.2.3. WAGO-I/O-Pro-CAA was installed from a CD into a folder named "Wago Software."

5.2.4. A folder named "_vbox_share_w" was used as a folder to interchange files between the Linux virtual environment and the Windows file system of the test PC.

5.2.5. A folder named "Wagotest" was used to hold subfolders for each individual device test.

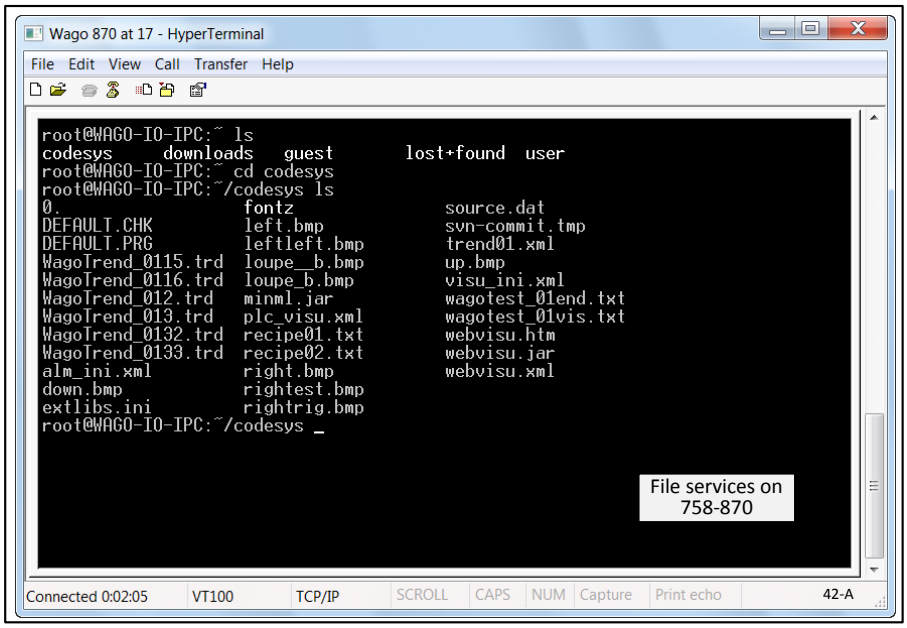
6. Basic File Services Testing

6.1. WAGO documentation indicates that the controllers operating with CoDeSys have operating systems that provide file services. Control programs prepared with CoDeSys can access these file services. The file system on the controllers was verified as follows:

6.1.1. The 758-870 IPC was fitted with an attached LCD display, keyboard, and mouse similar to that used on a standard personal computer. These devices allow the IPC to provide a Linux operator console in which commands can be entered, and the resulting displays can be viewed. This feature was confirmed by preforming simple tasks such as listing files, copying files, etc. This confirmed that the 758-870 IPC provided file services.

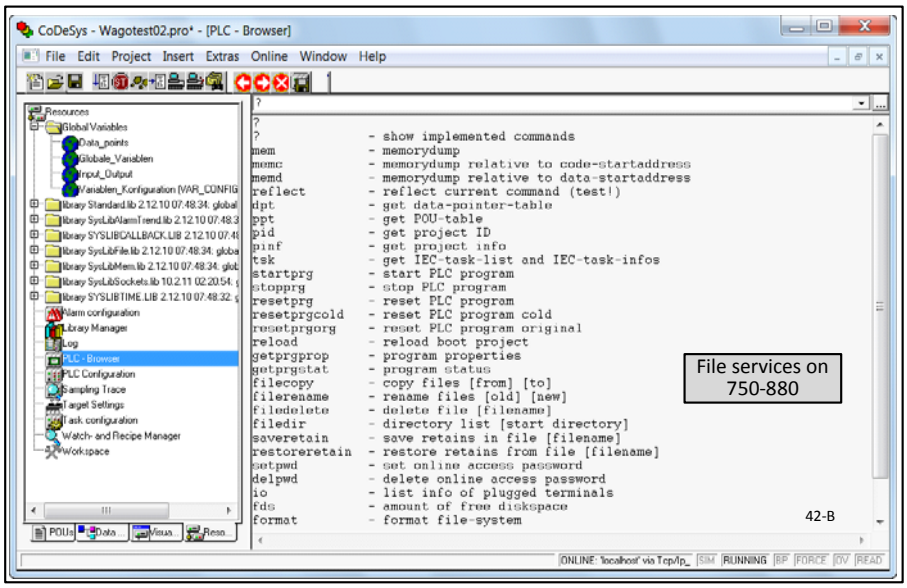
6.1.2. The 758-870 IPC uses the Linux operating system which does not normally use drive names (i.e. C:, E:, etc) as done with Windows computers. All attached storage devices are integrated into a single overall file system that begins at the "root" directory, noted as "\" in Linux. Examination of the 758-870 IPC file system revealed that an attached USB storage would be mounted at folder location "\media\sda1\". This storage location, although separate from the built-in file storage of the IPC, could be accessed in a manner to internal storage.

6.1.3. The test computer was also connected to the 758-870 IPC via FTP (File Transfer Protocol) using the local area network. The FTP client in Total Commander was used to transfer files to and from the IPC, further demonstrating file service capability. The same file structure accessible at the local Linux console was also accessible from the networked computer via FTP.



Operator console of 758-870 IPC

6.1.4. The 750-880 controller does not accommodate an attached keyboard and display. However, its file system could be accessed via FTP in a manner similar to the 758-870 IPC. Attached storage is provided on the 750-880 via an inserted SD Card (similar to a camera memory card). Using FTP, the attached storage was found to be installed as drive S (i.e. "S:") with access similar to that of a DOS/Windows disc drive. The 750-880 also supports access via the CoDeSys PLC Browser feature. That facility allowed direct access to the controller for simple file service operations as shown below:



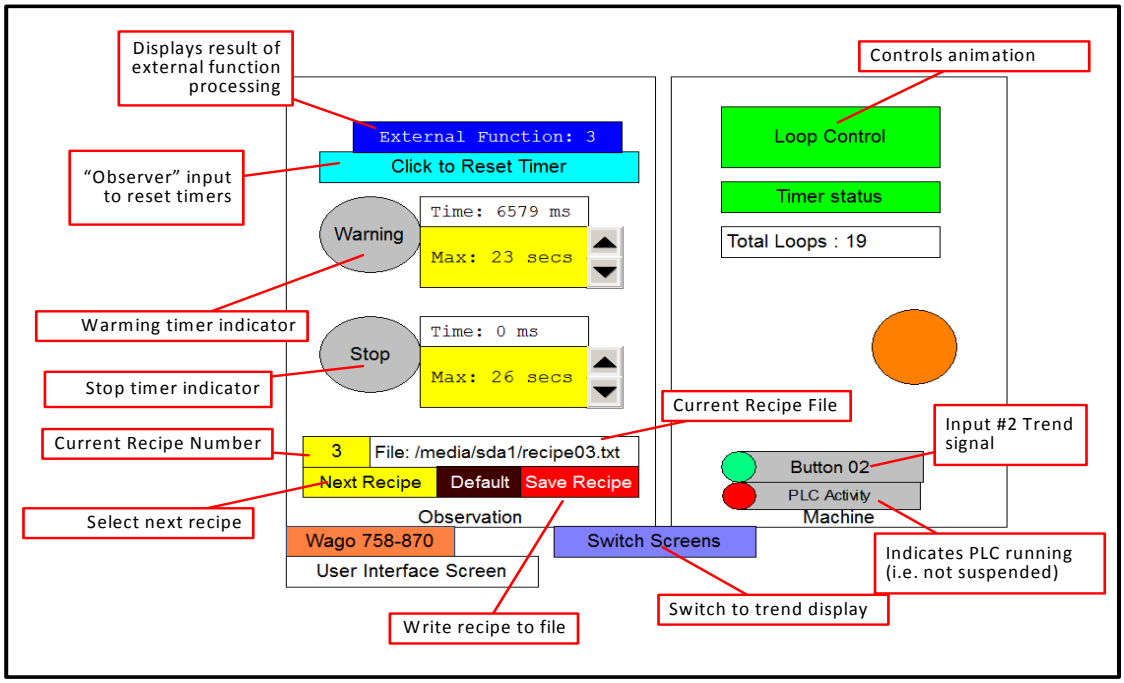
PLC Browser display of 750-880 Controller

7. File Services in Control Programs

7.1. A demonstration program was created to test a WAGO controller's ability to use the file services of the device's operating system. The program was based on the "first program" example provided in WAGO documentation. The test involved writing and reading "recipe" values to memory devices within and attached to the controller. This demonstrated storing and retrieving any form of PLC measured data, and potentially trend data, as provided for in a control program. A printout detailing the test program output from the CoDeSys program development software is attached later in this report.

7.2. A plain text editor in Total Commander was used to prepare four text files containing different recipes for values to preset Warning and Alarm timers within the demonstration program. The file names were recipexx.txt, with xx ranging from 01 to 04. The first two positions in the file are the warning time in seconds. A comma or space is in position 3, although any printing character can be used as a separator. Positions 4-5 contain the alarm time in seconds. For example, a text file containing "12,09" would indicate a warning time of 12 seconds, alarm time of 9 seconds. An ANSII/DOS standard line end (carriage return / line feed) was appended to the end of the line, although this is not necessary. Note that although timer preset values were used in the test, any PLC data value could be stored and retrieved in a similar manner.

7.3. A similar program was used to test both the 758-870 and 750-880 controllers using CoDeSys. The demo provides ability to modify, store, and load any of the 4 recipe files. The interface screen provides a button to cycle through the 4 files, and another button to use default timer settings. A screenshot of the program interface is shown below:



Demonstration Interface Screen for 758-870 IPC

7.4. The first two recipe files were stored in the default Codesys directory for each device. For the IPC, this directory is /home/codesys. No equivalent directory name was noted for the 750-880.

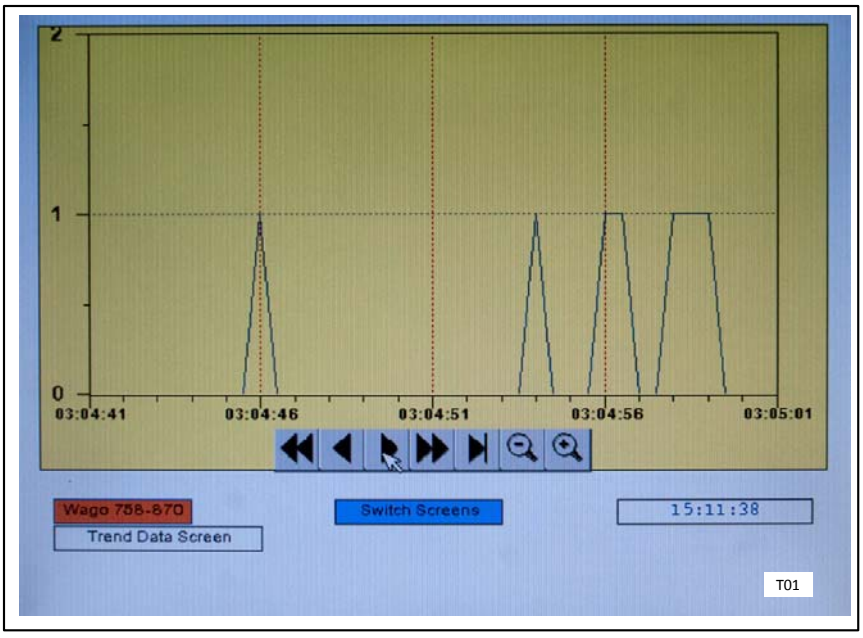
7.5. Recipe files three and four were stored on a memory device that was separate from the WAGO controller's internal storage. For the IPC 870, these files were stored on a USB drive mounted as directory /media/sda1. For the 750-880, the files were stored on the SD Card mounted as drive S:\.

7.6. The demo program's interface screen provides increase and decrease buttons for each of the current timer preset values in the PLC. The timers can be adjusted to a minimum of 5 and maximum of 99 seconds. A button is provided to save the current timer presets to the recipe file corresponding to the current recipe number. Another button is provided to load the corresponding recipe file. Therefore, editing recipe 3 and saving the settings will create a new file recipe03.txt.

7.7. The current file name is shown on the interface screen. The file will be saved to either internal or external storage depending on the recipe number. Recipes 1 and 2 go to internal storage, and recipes 3 and 4 go to the removable external (USB) storage. Pressing the load button preforms the reverse action, resulting in new values being loaded from storage.

8. Trend Data Test

8.1. The CoDeSys Visualization tend system was used to store the value of the second digital input (IEC address %IX0.1) to a file every 500 ms. The resulting data storage file was created by CoDeSys in the default directory on each unit. The contents of the current and stored log file could be seen on the trend screen.



Trend display from 758-870 IPC

9. Testing of File Service to Load User Routine

9.1. Section 12 of the technical manuals for Wago 758-870 IPC controllers describes the method of linking CoDeSys with external functions programmed in the C language. This method was followed to prepare and install a loadable user routine within the same demonstration program used for the data store and retrieve.

9.2. Using section 12 as a guide, the CoDeSys lib manager and associated ini file were set up to expect a lib named "libymtest.so" containing a single C-Function named MyTestFunction. The demo function accepts a single word parameter, modifies it, and returns it to CoDeSys. This forms an example of linking a loadable external function into a control program on a PLC.

9.3. Two similar external functions were prepared in C, named demo_function_x1 and demo_function_x5. The first program increments the passed word parameter by 1, and resets the count to 1 when it exceeds 10. The second program does the same action with an increment by 5 and max of 50, and reset to 0. The passed parameter is a simple counter representing the number of times the user has clicked a screen button to reset the warning and alarm timers. These two similar programs provided easy verification of which user routine was being used by the CoDeSys demo program. The text of each program is included later in this report.

9.4. Both programs were prepared and compiled with Ubuntu Linux running within a VirtualBox on a Windows 7 PC. A script file named makedemo.sh was prepared containing the proper gcc command line syntax. The compile process for each program produced a linkable library called libmytes.so.

9.5. After compiling each C file with makedemo, the resulting .so library was renamed to either libmytest_1x.so or libmytest_5x.so. This provided two similar library files that could be used for testing the IPC ability to link C library routines. The two library files were exported Linux environment to Windows using a shared folder. All testing after that required only the Windows environment.

9.6. Ax explained in section 12, a simple initialization file named "extlibs.ini" was prepared to direct loading of the user routine when the control program starts on the 758-870 IPC. In this simple test, the ini file directed the CoDeSys runtime system to load the single user routine contained in libmytest.so.

9.7. To test each routine, the FTP client in Total Commander on the PC was used to transfer the required files into the 758-880 IPC. The ini file was copied into directory /home/codesys within the Linux file system structure. The library to be tested was copied into directory /lib with the required rename of libmytest.so. Therefore, the contents of the user routine function library differed depending on which library file (_x1 or _x5) was copied to the IPC as libmytest.so.

9.8. Testing confirmed that CoDeSys running on the IPC used the file services of the operating system to load the designated library file containing the user routine. The file was loaded based on information placed within the CoDeSys control program as determined by the library manager on the development system, and by the associated ini file in the controller.

9.9. The two versions of the user routine were tested on the IPC simply by replacing the .so file in the proper directory on the IPC. No changes were required to the control program itself to accomplish a change in the user routine. Restarting the control program resulted in loading of the then-current user routine (i.e. _1X or _5X).

9.10. When CoDeSys first loads, it apparently links and loads external library functions as directed by the "extlibs.ini" file. After loading, it does not seem to need or reference external lib files. A library can be deleted without affecting a running CoDeSys. Rebooting CoDeSys without the expected library causes it to reuse the last version of the library functions. But, if a different libmytest.so library file is copied to the IPC /lib directory, CoDeSys will restart with whatever library functions are contained in /lib/libmytest.so.

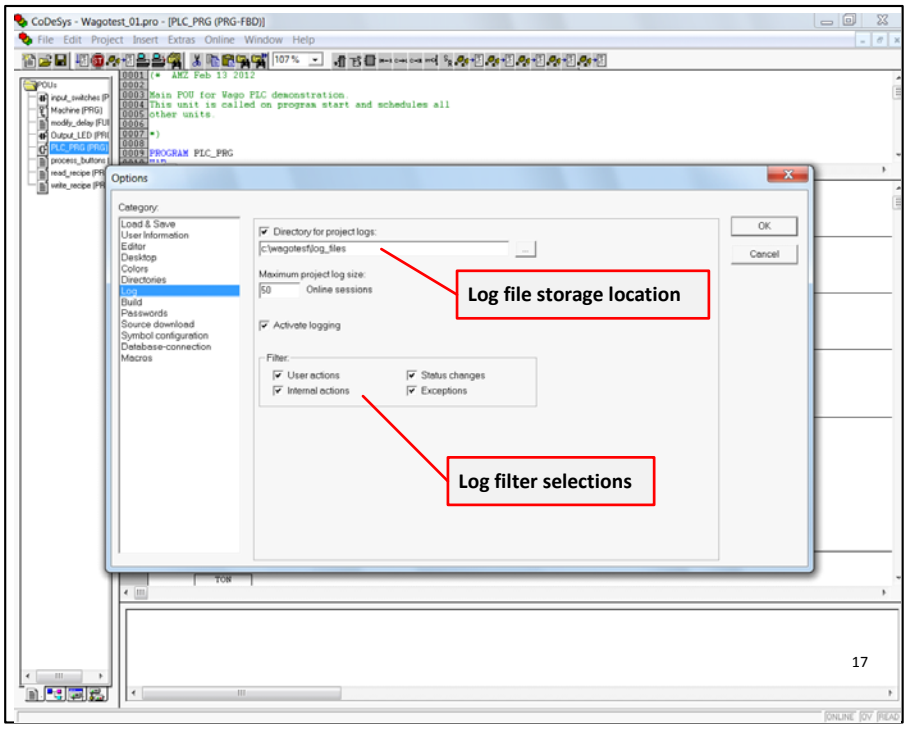
9.11. The initialization file extlibs.ini must be present and correct when CoDeSys starts. It can be deleted while the program is running. Restarting without extlibs.ini in place causes the system to hang. Also, the file cannot be copied to the directory path /home/codesys while the program is trying to start (the linux file handle is likely locked). The linux console must be used to kill the task associated with the CoDeSys runtime system (plclinux_rt) before the extlibs file can be recopied into the /home/codesys folder. Details of this is provided in section 12 of the WAGO 758-870 manual. The CoDeSys program plclinux_rt can then be restarted at the command line, or by a system reset.

10. Testing of Audit System

10.1. The CoDeSys interaction tracking system was tested during development and testing of the file service and external function tests. The process consisted of the following steps:

10.1.1. Enabling the Log feature within CoDeSys via a check mark in a dialog box.

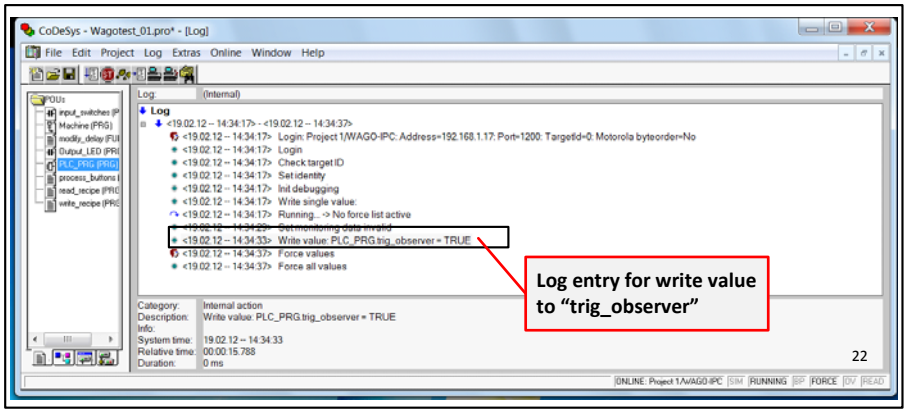
10.1.2. Specifying the location of the log file for the current project. This included storing log data on the development computer as well to a networked storage device.



CoDeSys Log setup dialog box

10.1.3. Performing online actions such as logging in to a controller, forcing data values, and starting/stopping the PLC program within the controller.

10.1.4. Noting the log entries that track user interactions for different online sessions.



CoDeSys Log display

10.2. The observed action of the CoDeSys log system matched that expected based on the related CoDeSys documentation.

11. Testing of Step Mode

11.1. The online debugging features of WAGO-I/O-PRO CAA were tested while developing the demonstration program detailed elsewhere in this report. The tested features include:

11.1.1. Using the development system (access tool) to install one or more breakpoints within the PLC program on the controller.

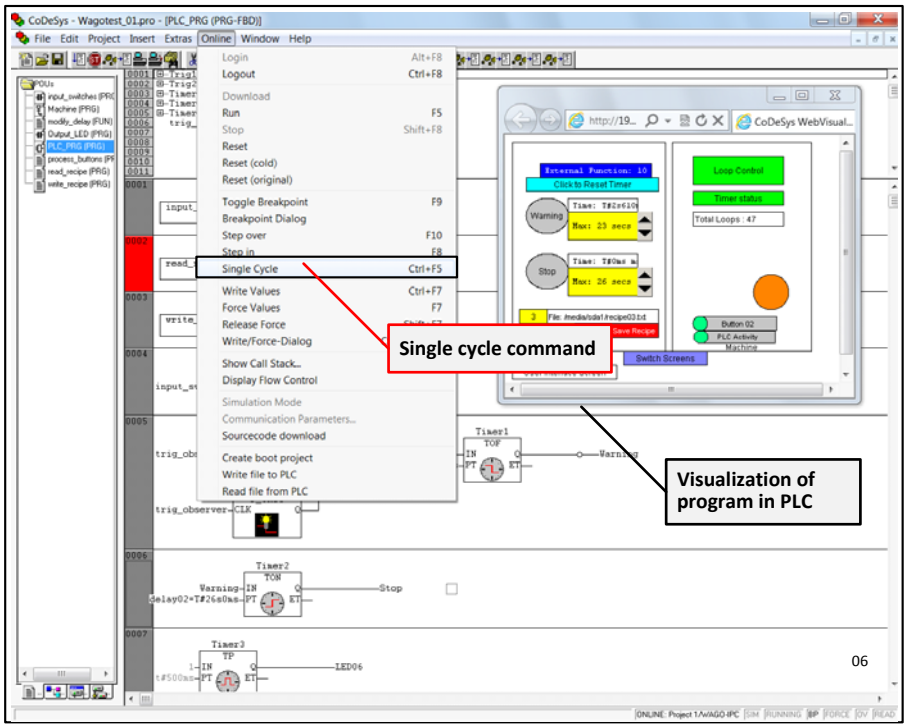
11.1.2. Using the access tool to start and stop program operation with and without breakpoints installed in the controller.

11.1.3. Observing suspension of an operating program at a breakpoint as noted by an onscreen indicator that blinks only when the program is running in the PLC.

11.1.4. Using the access tool to observe and change PLC data while the program was suspended in the PLC.

11.1.5. Using the access tool to initiate and observe single cycle operation from a single breakpoint. This was noted by incremental motion of the on-screen animation that normally runs without stopping.

11.1.6. Observing single step operation in a controller that had suspended operation.



CoDeSys Online Menu

12. Demonstration Program Code

12.1. The C program used for the first demonstration C-Function is demo_function_1x:

```
/*  Wago C-Function demonstration  AMZ Feb 2012
demo_function_1x
```

```
increment passed value by 1 with max of 10.
Reset to 1.
*/

#include <stdio.h>

#define max_val 10  /* max counter will increase in this sub */

unsigned short cur_val;
unsigned short MyTestFunction(unsigned short value)
{
cur_val = value + 1; /* increment current value */
if (cur_val > max_val) cur_val = 1;
return cur_val;
}
```

12.2. The second demonstration function is C program demo_function_5X

```
/* Wago C-Function demonstration  AMZ Feb 2012
demo_function_5x
increment passed value by 5 with max of 50.
Reset to 0.

*/

#include <stdio.h>

#define max_val 50  /* max counter will increase in this sub */

unsigned short cur_val;
unsigned short MyTestFunction(unsigned short value)
{
cur_val = value + 5; /* increment current value */
if (cur_val > max_val) cur_val = 0;
return cur_val;
}
```

12.3. The script file used to compile the demo programs is makedemo.sh:

```
echo this is a script made with gedit under linux
gcc demo_function.c -Wall -shared -o libmytest.so
echo compile complete.
```

12.4. The CoDeSys initialization file extlibs.ini was programmed as follows:

```
[EXT_LIB_LIST]
```

```
1=mytest

[mytest]
1=MyTestFunction
```

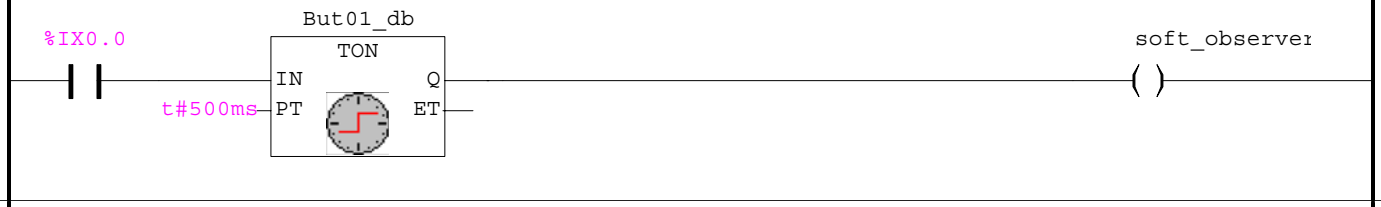
13. WAGO-I/O-PRO-CAA Project Files

13.1. The CoDeSys program development program used in WAGO-I/O-PRO-CAA can produce a printout for project content such as control program listings, I/O assignments, internal variables, etc. This feature was used to produce printouts of project information for the 758-870 and 750-880 controllers. A complete project printout can be very long and detailed, and contains details not relevant to this test. Therefore, not all project content was included in the reports that follow.

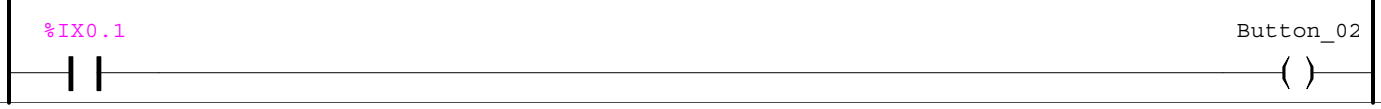
Filename:	Wagotest_01.pro
Directory:	C:\Wagotest\Wagotest_758
Change date:	20.2.12 17:15:51 / V2.3
Title:	Wago PLC demonstration
Author:	Arthur Zatarain
Version:	1.0
Description:	Demo package in Wago Pro CAA.

```
0001  (* AMZ Feb 13 2012
0002
0003  Demo program to read physical input switches to activate
0004  soft equivalent within program.
0005  *)
0006
0007  PROGRAM input_switches
0008  VAR
0009      Button_02: BOOL;
0010      But01_db: TON;
0011      soft_observer: BOOL;
0012      But02_db: TON;
0013  END_VAR
```

0001

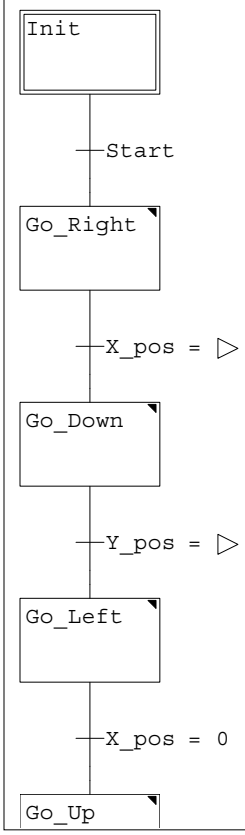


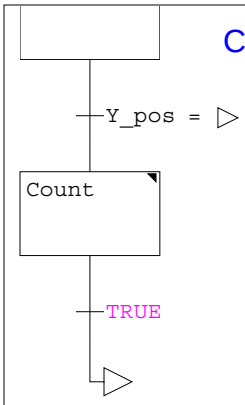
0002



Machine (PRG-SFC)

```
0001  (* AMZ Feb 13 2012
0002
0003  Demonstration of screen animation adopted from Wago/CoDeSys demo.
0004  *)
0005
0006  PROGRAM Machine
0007  VAR
0008      X_pos: INT;
0009      Y_pos: INT;
0010      Counter: INT;
0011  END_VAR
```





Machine (PRG-SFC).Aktion Go_Right (ST)

```
0001 X_pos := X_pos + 1 ;
```

Machine (PRG-SFC).Action Go_Down (ST)

```
0001 Y_pos := Y_pos + 1 ;
```

Machine (PRG-SFC).Aktion Go_Left (ST)

```
0001 X_pos := X_pos - 1 ;
```

Machine (PRG-SFC).Action Go_Up (ST)

```
0001 Y_pos := Y_pos - 1 ;
```

Machine (PRG-SFC).Aktion Count (ST)

```
0001 Counter := Counter + 1 ;
0002
0003
0004
0005
0006
```

modify_delay (FUN-ST)

```
0001 (* AMZ Feb 13 2012
0002
0003 Demo program to process screen buttons to modify alarm and warning
0004 timer values.
0005 *)
0006
0007 FUNCTION modify_delay : BOOL
0008 VAR_INPUT
0009   delay_number : WORD;  (* which delay to modify *)
0010   delay_change : INT; (* variance *)
0011 END_VAR
0012 VAR
0013   tempw : WORD;
0014   temp_delay : INT;
0015   temp_secs : WORD;
0016
0017 END_VAR
```

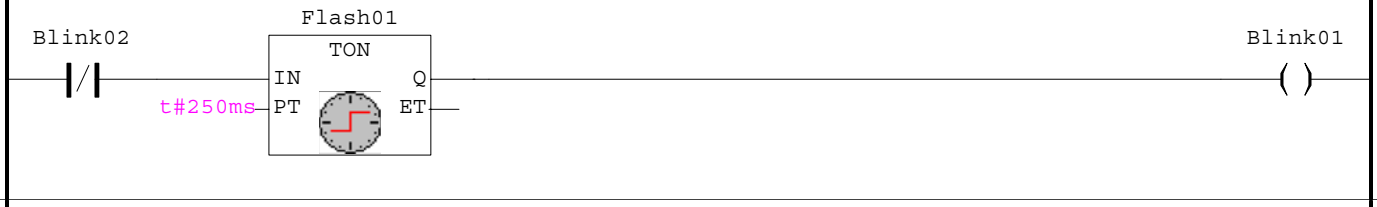
```
0001 (* change delay time in seconds based on passed incremental value *)
0002
0003 modify_delay := FALSE;
0004
0005 CASE delay_number OF
0006     1 : temp_delay := TIME_TO_INT(delay01);
0007     2 : temp_delay := TIME_TO_INT(delay02);
0008 END_CASE
0009
0010 temp_delay := temp_delay / 1000;  (* convert ms to seconds *)
0011 temp_delay := temp_delay + delay_change;  (* convert in whole seconds *)
0012
0013 IF temp_delay < 5 THEN temp_delay := 5; END_IF
0014 IF temp_delay > 99 THEN temp_delay := 99; END_IF
0015 temp_secs := temp_delay;  (* delay in seconds *)
0016
0017 temp_delay := temp_delay * 1000;  (* back to ms *)
0018
```

```
0019 CASE delay_number OF
0020   1 : delay01 := INT_TO_TIME(temp_delay); delay01_secs := temp_secs;
0021   2 : delay02 := INT_TO_TIME(temp_delay); delay02_secs := temp_secs;
0022 END_CASE
0023
0024 modify_delay := TRUE;  (* return valu *)
0025
```

Output_LED (PRG-LD)

```
0001 (* AMZ Feb 13 2012
0002
0003 Ladder logic to provide global flash timer.
0004 *)
0005
0006 PROGRAM Output_LED
0007 VAR
0008     Flash01: TON;
0009     Flash02: TON;
0010     Blink01: BOOL;
0011     Blink02: BOOL;
0012 END_VAR
```

0001



0002

Bool to control flashing

0003

0004

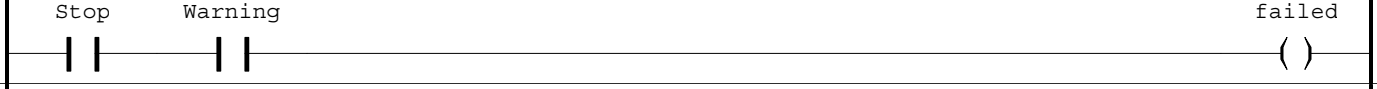
Normal run light

0005

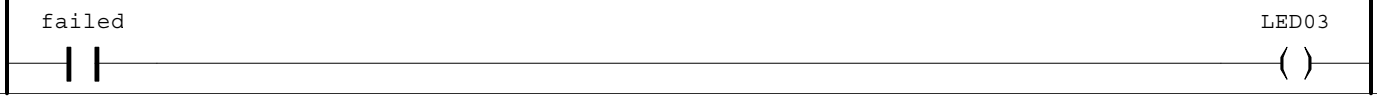
Warning lightt

0006

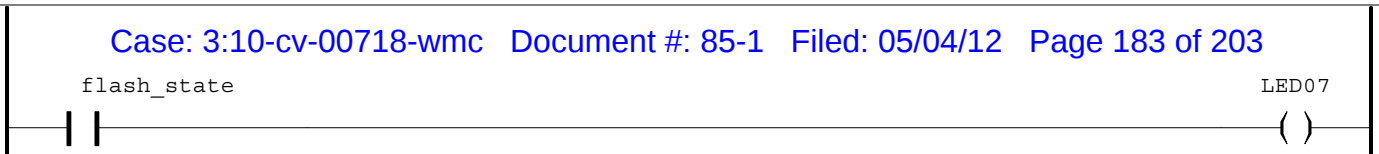
Failed light



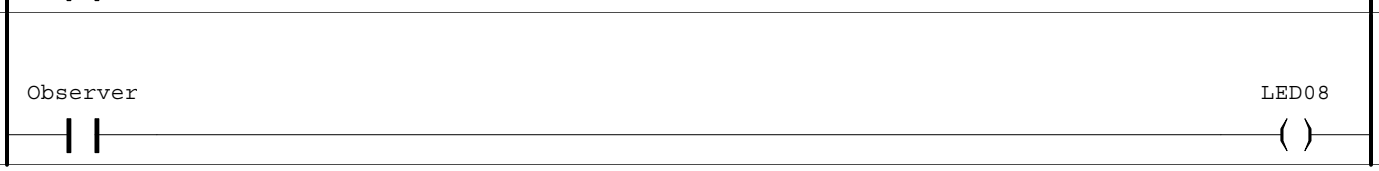
0007



0008



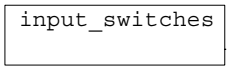
0009



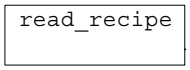
PLC_PRG (PRG-FBD)

```
0001 (*  AMZ Feb 13 2012
0002
0003 Main POU for Wago PLC demonstration.
0004 This unit is called on program start and schedules all
0005 other units.
0006
0007 *)
0008
0009 PROGRAM PLC_PRG
0010 VAR
0011     Trig1: R_TRIG;
0012     Trig2: F_TRIG;
0013     Timer1: TOF;
0014     Timer2: TON;
0015     Timer3: TP;
0016     trig_observer: BOOL;
0017 END_VAR
```

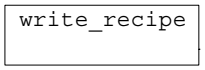
0001



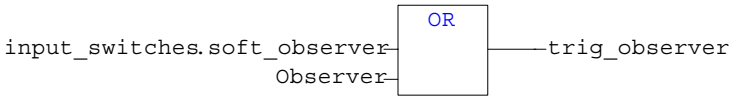
0002



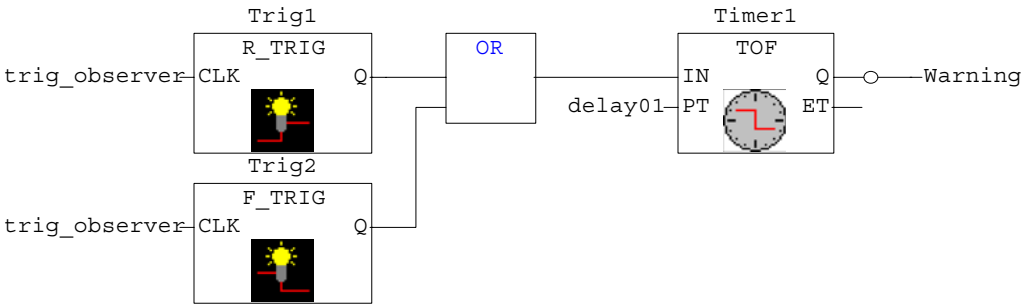
0003



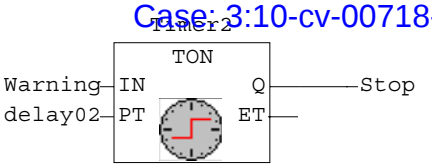
0004



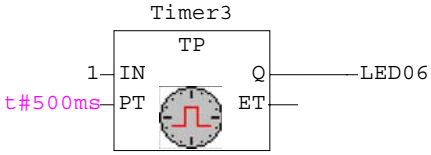
0005



0006



0007



0008

Update the PLC indiator LEDs

Output_LED

0009

Proces the functions related to button presses

process_buttons

0010

do not continue if stopped

Stop Return

0011

Operate state machine when not stopped.

Machine

process_buttons (PRG-ST)

```
0001 PROGRAM process_buttons
0002 VAR
0003
0004 END_VAR
```

```
0001 (*  AMZ Feb 13 2012
0002     demo program button processor *)
0003
0004 (* increment passed value on first transition of the
0005 variable observer. The increment will depend on which linkable
0006 routine is present on the controller during runtime.
0007
0008 All versions of the demo function are called by the same
0009 function name MyTestFunction
0010 *)
0011
0012 IF observer   THEN
0013   demo_val := MyTestFunction(demo_val);
0014 (*  count01 := TRUE; *)
0015   observer := FALSE;
0016 END_IF;
0017
0018
0019 (* process buttons for delay value change. *)
0020
0021 IF del01u THEN
0022   modify_delay(1, 1);  (* timer 1 up 1 *)
0023   del01u := FALSE;
```

```
0024 END_IF
0025
0026 IF del01d THEN
0027   modify_delay(1, -1);  (* timer 1 up 1 *)
0028   del01d := FALSE;
0029 END_IF
0030
0031 IF del02u THEN
0032   modify_delay(2, 1);  (* timer 1 up 1 *)
0033   del02u := FALSE;
0034 END_IF
0035
0036 IF del02d THEN
0037   modify_delay(2, -1);  (* timer 1 up 1 *)
0038   del02d := FALSE;
0039 END_IF
```

read_recipe (PRG-ST)

```
0001 PROGRAM read_recipe
0002 VAR
0003     fname : STRING := 'none';
0004     infile: DWORD;
0005     inbuffer: ARRAY[0..9] OF BYTE;
0006     readbytes: DWORD := 5;   (* read only 5 bytes, 2 values sep by comma or space *)
0007     file_read_count: DWORD;
0008     file_close_err : BOOL;
0009     tempw: WORD;
0010     new_recipe : INT;
0011 END_VAR
0012 VAR_OUTPUT
0013 END_VAR
```

```
0001 (* AMZ Feb 13 2012
0002 Demonstration of Wago PLC program to read recipe information from a text file
0003 using the file services on the PLC.
0004
0005 Four files can be read as determined by variable cur_recipe.
0006
0007 The text file has 5 signifiant characters. First 2 are ascii decimal of
0008 warning delay.Then a space (or comma), then the stop delay.
0009 Minimum delay is 5 seconds, max is 99.
0010 *)
0011
0012 IF read_recipe_button THEN
0013       IF curr_recipe < 1 THEN
0014         new_recipe := 1;  (* try to load the first recipe file *)
0015       ELSE
0016         new_recipe := curr_recipe+1; (* next recipe file *)
0017       END_IF
0018      IF new_recipe > 4 THEN new_recipe := 1; END_IF  (* max of 4 recipes *)
0019
0020     CASE new_recipe OF
0021         1: fname := rec_fname_01;
0022         2: fname := rec_fname_02;
0023         3: fname := rec_fname_03;
0024         4: fname := rec_fname_04;
0025         ELSE fname := rec_fname_01;
0026     END_CASE
0027
0028     infile := SysFileOpen(fname, 'r');  (* try to open the file *)
0029     file_read_count := 0;  (* assume no bytes will be read *)
0030     IF (infile >0) THEN  (* file opend ok *)
0031         file_read_count := SysFileRead(infile, ADR(inbuffer), readbytes);
0032         file_close_err := SysFileClose(infile);
0033     ELSE
0034          load_def_recipe := TRUE;  (* set defaults if no file read *)
0035       END_IF
0036
0037       IF (file_read_count >= readbytes) THEN    (* convert first 2 characters in the
0038                                                 array to delay0 word *)
0039 (* remove ascii offset 48 *)
0040         tempw := ((inbuffer[0] - 48) * 10) + (inbuffer[1] - 48);
```

```
0041        IF tempw < 5 THEN tempw := 5; END_IF
0042        delay01_secs := tempw;
0043        tempw := tempw * 1000;  (* convert to milliseconds *)
0044        delay01 := WORD_TO_TIME(tempw);  (* warming timer *)
0045    (* skip over separator at [2] and get next 2 digits *)
0046        tempw := ((inbuffer[3] -48) * 10) + (inbuffer[4] - 48);
0047        IF tempw < 5 THEN tempw := 5; END_IF
0048        delay02_secs := tempw;
0049        tempw := tempw * 1000;  (* convert to milliseconds *)
0050        delay02 := WORD_TO_TIME(tempw);  (* stop timer *)
0051        curr_recipe := New_recipe;
0052    END_IF
0053
0054    read_recipe_button := FALSE;
0055 END_IF
0056
0057 IF load_def_recipe THEN
0058    new_recipe := 0;  curr_recipe:= 0;  (* use defaults *)
0059    fname := 'None - using default';
0060    delay01 := default_delay; delay02 := default_delay;
0061    delay01_secs := TIME_TO_WORD(default_delay / 1000);  (* delay in seconds *)
0062    delay02_secs := delay01_secs;
0063    load_def_recipe := FALSE;
0064    read_recipe_button := FALSE;
0065 END_IF
0066
```

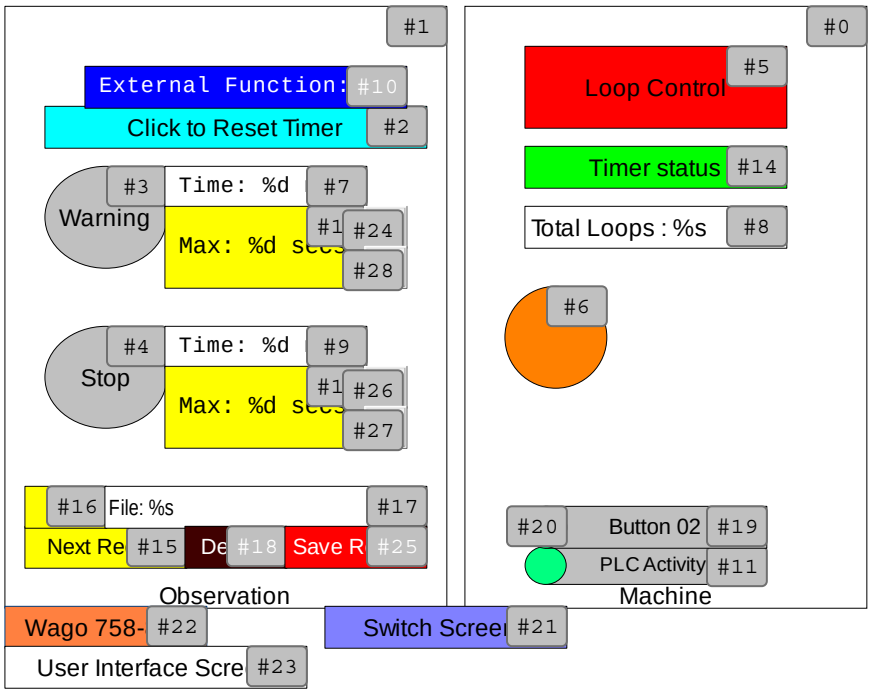
write_recipe (PRG-ST)

```
0001 PROGRAM write_recipe
0002 VAR
0003    fname : STRING := 'none';
0004    outfile: DWORD;
0005    outbuffer: ARRAY[0..9] OF BYTE;
0006    writebytes: DWORD := 4;   (* read only 4 bytes, 2 words *)
0007    file_write_count: DWORD;
0008    file_close_err : BOOL;
0009    tempw: WORD;
0010    save_recipe : INT;
0011    ii : WORD;
0012 END_VAR
0013 VAR_OUTPUT
0014 END_VAR
```

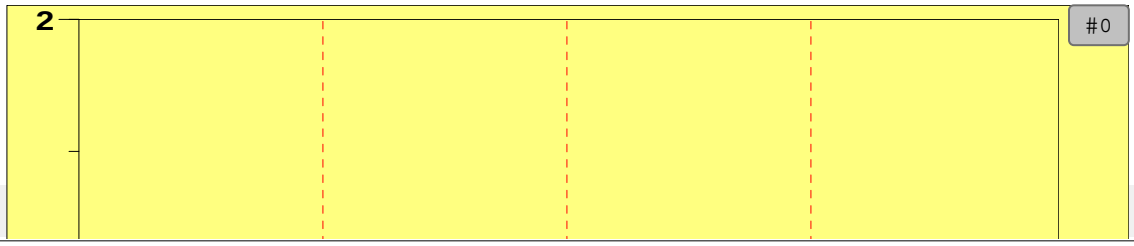
```
0001 (* AMZ Feb 13 2012
0002 demonstration of Wago PLC program to write date into a file using
0003 the file services on the PLC. The files to be written is determined by
0004 variable curr_recipe. The file has two 2-characer values in a simple
0005 text file in the format ABxCD.  AB is the 2 character representation
0006 of the warning delay in seconds. The middle character x can be anything,
0007 and a space is used in this example.
0008 The final pair CD is the 2 char representation of the alarm time in seconds.
0009 *)
0010
0011 IF write_recipe_button THEN
0012    save_recipe := curr_recipe; (* local of current global recipe *)
0013    IF ((save_recipe <1) OR (save_recipe > 4)) THEN
0014        load_def_recipe := TRUE;
0015        read_recipe; (* use default values *)
0016        write_recipe_button := FALSE;
0017    END_IF
0018 END_IF
0019
0020 IF write_recipe_button THEN  (* still OK to write *)
0021    save_recipe := curr_recipe;  (* max of 4 recipes *)
0022    CASE save_recipe OF
0023        1: fname := rec_fname_01;
0024        2: fname := rec_fname_02;
0025        3: fname := rec_fname_03;
0026        4: fname := rec_fname_04;
0027    END_CASE
0028
0029 (* put space characters into the array to delay0 word *)
```

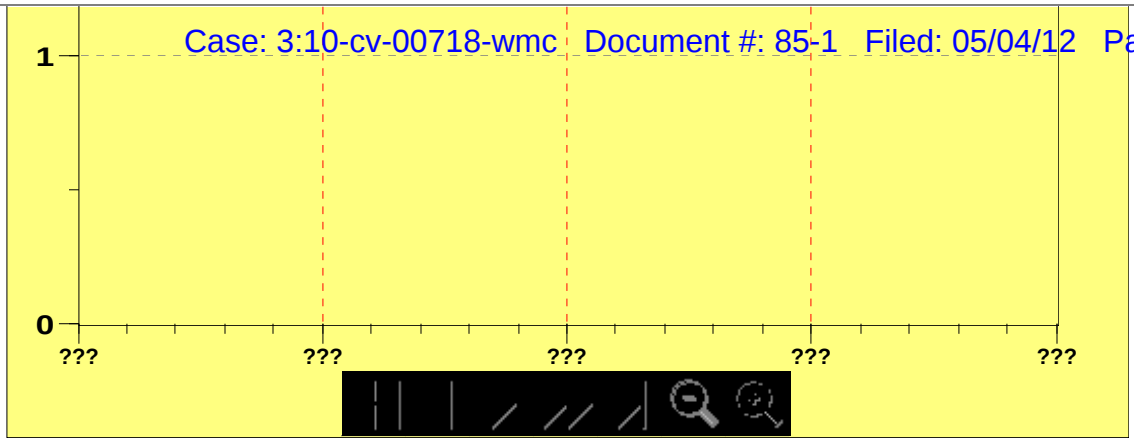
```
0030      FOR ii := 0 TO 9 DO
0031        outbuffer[ii] := 32;  (* ascii space *)
0032      END_FOR
0033      tempw := TIME_TO_WORD(delay01) / 1000; (* milliseconds to secs *)
0034      IF tempw < 5 THEN tempw := 5; END_IF  (* min time allowed *)
0035      IF tempw > 99 THEN tempw := 99;  END_IF (* max delay secs *)
0036      IF tempw > 9 THEN  (* need 2 digits *)
0037        outbuffer[0] := WORD_TO_BYTE(tempw / 10) + 48;  (* upper digit *)
0038      ELSE
0039        outbuffer[0] := 48;  (* upper digit is ascii 0*)
0040      END_IF
0041      outbuffer[1] := WORD_TO_BYTE(tempw MOD 10) + 48;  (* lower *)
0042
0043      tempw := TIME_TO_WORD(delay02) / 1000;
0044      IF tempw > 99 THEN tempw := 99; END_IF
0045      IF tempw > 9 THEN
0046        outbuffer[3] := WORD_TO_BYTE(tempw / 10) + 48;  (* upper digit *)
0047      ELSE
0048        outbuffer[3] := 48;  (* upper digit is ascii 0*)
0049      END_IF
0050      outbuffer[4] := WORD_TO_BYTE(tempw MOD 10) + 48;  (* lower *)
0051      outbuffer[5] := 16#0A; outbuffer[6] := 16#0D;  (* cr/lf pair *)
0052
0053  (* write buffer to file *)
0054      outfile := SysFileOpen(fname, 'w');  (* try to open the file *)
0055      writebytes := 10;  (* write all 10 bytes *)
0056      IF (outfile > 0) THEN  (* file opend ok *)
0057          file_write_count := SysFileWrite(outfile, ADR(outbuffer), writebytes);
0058          file_close_err := SysFileClose(outfile);
0059      END_IF
0060      write_recipe_button := FALSE;
0061  END_IF
0062
```

PLC_VISU



trend01





Wago 758- #4 | Switch Scree #1 | %t%H:%M:%S #2
Trend Data Scree #3

Data_points

```
0001
0002  VAR_GLOBAL
0003  END_VAR
0004
0005  VAR_GLOBAL CONSTANT
0006  END_VAR
0007
0008  VAR_GLOBAL RETAIN
0009  END_VAR
0010
0011  VAR_GLOBAL RETAIN
0012  END_VAR
```

Globale_Variablen

```
0001  VAR_GLOBAL
0002      Observer : BOOL;
0003      Warning : BOOL;
0004      Stop : BOOL;
0005      Start : BOOL;
0006      flash_state : BOOL;
0007      demo_val : WORD;
0008    (*  count01 : BOOL;  *)
0009      default_delay : TIME := t#6000ms;
0010        delay01: TIME := t#6000ms;
0011      delay02: TIME := t#6000ms;
0012      delay01_secs, delay02_secs : WORD := 6; (* delay in whole seconds *)
0013
0014      failed: BOOL;
0015      read_recipe_button : BOOL;
0016      write_recipe_Button : BOOL;
0017      recipe_01 : BOOL;
0018      curr_recipe : INT :=0; (* current recipe in use, start with defaults *)
0019
0020  (* note: file names are platform specific *)
0021
0022      rec_fname_01 : STRING := 'recipe01.txt';
0023      rec_fname_02 : STRING := 'recipe02.txt';
0024      rec_fname_03 : STRING := '/media/sda1/recipe03.txt';
0025      rec_fname_04 : STRING := '/media/sda1/recipe04.txt';
0026
0027      load_def_recipe : BOOL := FALSE; (* this can be set globally *)
0028
0029      change_visu_button : BOOL := FALSE;
0030      soft_sel_visu : BOOL := FALSE;
0031      cur_visu: WORD := 1;
0032      new_visu: WORD := 1;
0033
0034      del01u : BOOL := FALSE;  (* button for delay01 up *)
0035      del01d : BOOL := FALSE; (* button for delay01 down *)
```

```
0036      del02u : BOOL := FALSE;  (* button for delay02 up *)
0037      del02d : BOOL := FALSE; (* button for delay02 down *)
0038  END_VAR
```

Input_Output

```
0001
0002  VAR_GLOBAL
0003  END_VAR
0004
0005  VAR_GLOBAL CONSTANT
0006  END_VAR
0007
0008  VAR_GLOBAL RETAIN
0009  END_VAR
```

Variablen_Konfiguration

```
0001  VAR_CONFIG
0002  END_VAR
```

Global_Variables

```
0001  VAR_GLOBAL
0002  END_VAR
```

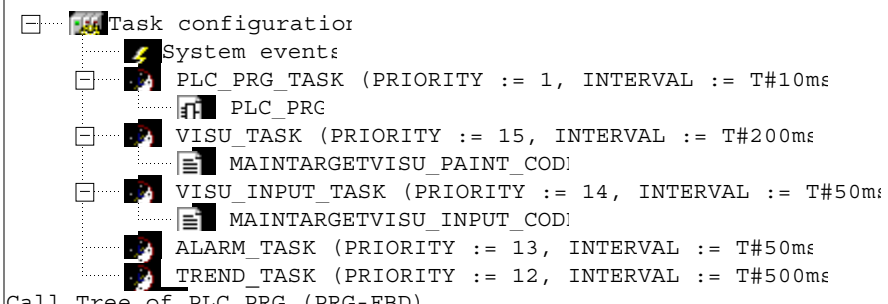
Globale_Variablen

```
0001  VAR_GLOBAL
0002  END_VAR
```

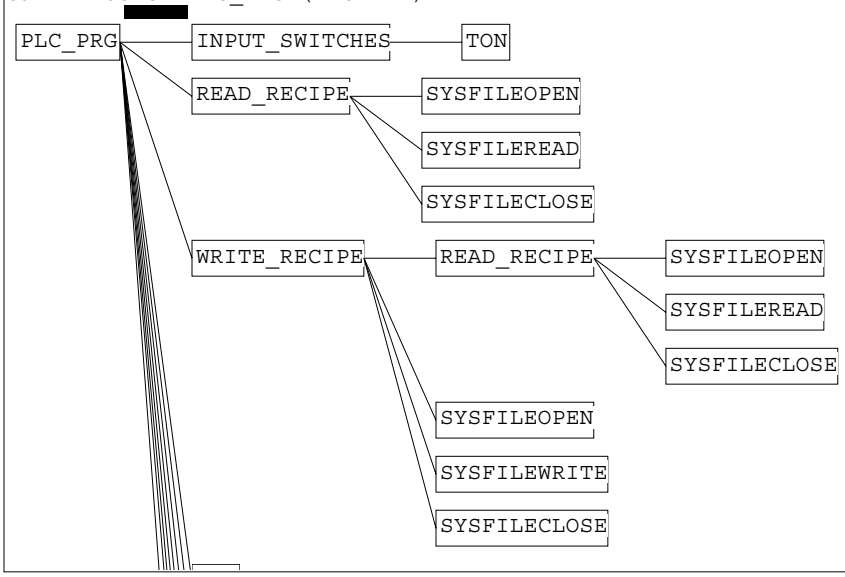
PLC Configuration

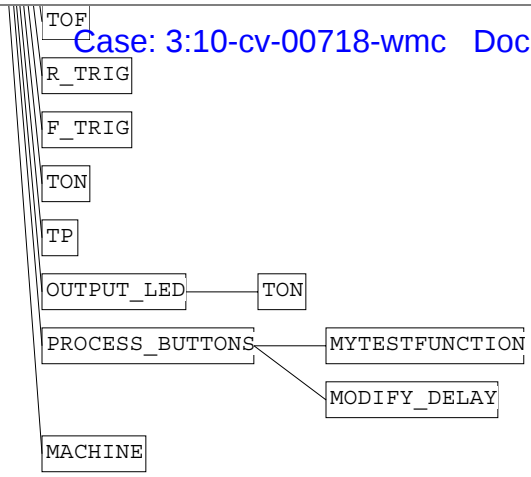
```
*PLC Configuration (Id.: 8765)
  Node number:  -1
  Input address: %IB0
  Output address: %QB0
  Diagnostic address: %MB0
  Download:  1
  AutoAdr:  1

  *K-Bus[FIX] (Id.: 11994)
    Node number:  0
    Input address: %IB0
    Output address: %QB0
    Diagnostic address: %MB0
    Download:  1
    AutoAdr:  1

    *0750-1502 8DI / 8DO 24V DC 0.5A Ribbon Cable[VAR] (Id.: 2000034008)
      Node number:  0
      Input address: %IB0
      Output address: %QB0
      Diagnostic address: %MB0
      Download:  1
      AutoAdr:  1
      Channels:
         AT %IX0.0: BOOL; (* Ch_1 Digital input *) [CHANNEL (I)]
        LED01 AT %QX0.0: BOOL; (* First output *) [CHANNEL (Q)]
         AT %IX0.1: BOOL; (* Ch_2 Digital input *) [CHANNEL (I)]
        LED02 AT %QX0.1: BOOL; (* Ch_2 Digital output *) [CHANNEL (Q)]
         AT %IX0.2: BOOL; (* Ch_3 Digital input *) [CHANNEL (I)]
        LED03 AT %QX0.2: BOOL; (* Ch_3 Digital output *) [CHANNEL (Q)]
         AT %IX0.3: BOOL; (* Ch_4 Digital input *) [CHANNEL (I)]
        LED04 AT %QX0.3: BOOL; (* Ch_4 Digital output *) [CHANNEL (Q)]
         AT %IX0.4: BOOL; (* Ch_5 Digital input *) [CHANNEL (I)]
        LED05 AT %QX0.4: BOOL; (* Ch_5 Digital output *) [CHANNEL (Q)]
         AT %IX0.5: BOOL; (* Ch_6 Digital input *) [CHANNEL (I)]
        LED06 AT %QX0.5: BOOL; (* Ch_6 Digital output *) [CHANNEL (Q)]
         AT %IX0.6: BOOL; (* Ch_7 Digital input *) [CHANNEL (I)]
        LED07 AT %QX0.6: BOOL; (* Ch_7 Digital output *) [CHANNEL (Q)]
         AT %IX0.7: BOOL; (* Ch_8 Digital input *) [CHANNEL (I)]
        LED08 AT %QX0.7: BOOL; (* Ch_8 Digital output *) [CHANNEL (Q)]
```

```
  *Internal Digital I/O[FIX] (Id.: 9)
    Node number:  1
    Input address: %IB512
    Output address: %QB512
    Diagnostic address: %MB0
    Download:  1
    AutoAdr:  1

    *Internal Digital Inputs[FIX] (Id.: 10)
      Node number:  0
      Input address: %IB512
      Output address: %QB512
      Diagnostic address: %MB4
      Download:  1
      AutoAdr:  1
      Channels:
        DIN1 AT %IX2300.0: BOOL; (* Internal Digital Input Bit 1 *) [CHANNEL (I)]
        DIN2 AT %IX2300.1: BOOL; (* Internal Digital Input Bit 2 *) [CHANNEL (I)]

    *Internal Digital Outputs[FIX] (Id.: 11)
      Node number:  1
      Input address: %IB513
      Output address: %QB512
      Diagnostic address: %MB8
      Download:  1
      AutoAdr:  1
      Channels:
        DOUT1 AT %QX2300.0: BOOL; (* Internal Digital Output Bit 1 *) [CHANNEL (Q)]
        DOUT2 AT %QX2300.1: BOOL; (* Internal Digital Output Bit 2 *) [CHANNEL (Q)]

  *Fieldbus variables[FIX] (Id.: 2010310001)
    Node number:  3
    Input address: %IB0
    Output address: %QB0
    Diagnostic address: %MB0
    Download:  1
    AutoAdr:  1
Task configuration
```



Call Tree of PLC_PRG (PRG-FBD)

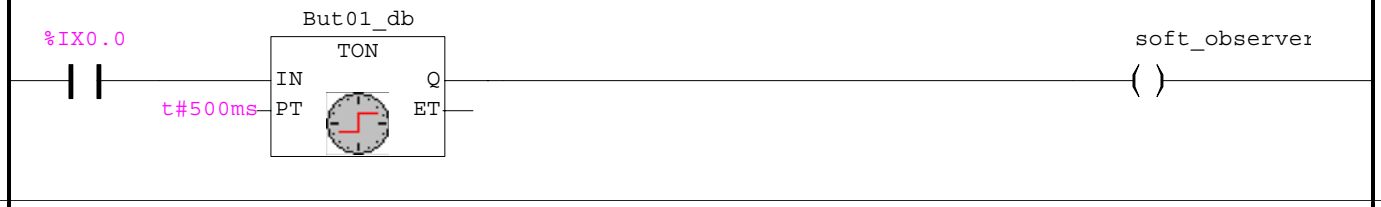



TOF
R_TRIG
F_TRIG
TON
TP
OUTPUT_LED
TON
PROCESS_BUTTONS
MYTESTFUNCTION
MODIFY_DELAY
MACHINE

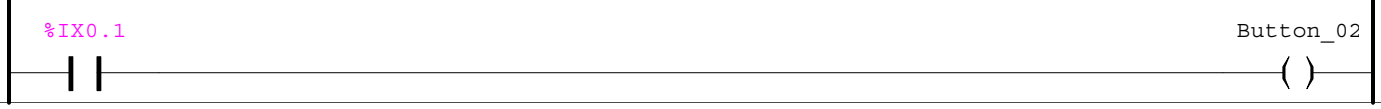
Filename:	Wagotest02.pro
Directory:	C:\Wagotest\Wagotest_750
Change date:	20.2.12 17:20:51 / V2.3
Title:	Wago PLC demonstration
Author:	Arthur Zatarain
Version:	1.0
Description:	Demo package in Wago Pro CAA.

```
0001  (* AMZ Feb 13 2012
0002
0003  Demo program to read physical input switches to activate
0004  soft equivalent within program.
0005  *)
0006
0007  PROGRAM input_switches
0008  VAR
0009      Button_02: BOOL;
0010      But01_db: TON;
0011      soft_observer: BOOL;
0012      But02_db: TON;
0013  END_VAR
```

0001

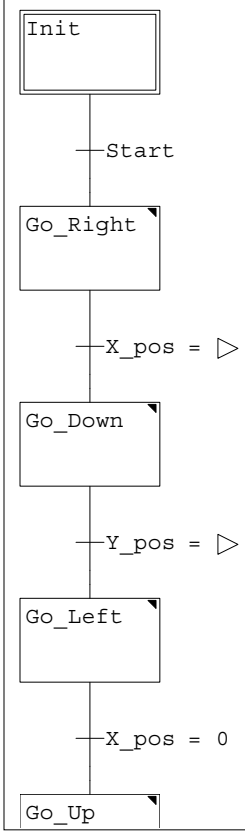


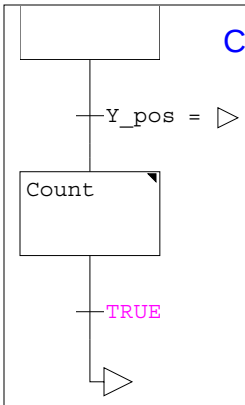
0002



Machine (PRG-SFC)

```
0001  (* AMZ Feb 13 2012
0002
0003  Demonstration of screen animation adopted from Wago/CoDeSys demo.
0004  *)
0005
0006  PROGRAM Machine
0007  VAR
0008      X_pos: INT;
0009      Y_pos: INT;
0010      Counter: INT;
0011  END_VAR
```





Machine (PRG-SFC).Aktion Go_Right (ST)

```
0001  X_pos := X_pos + 1 ;
```

Machine (PRG-SFC).Action Go_Down (ST)

```
0001  Y_pos := Y_pos + 1 ;
```

Machine (PRG-SFC).Aktion Go_Left (ST)

```
0001  X_pos := X_pos - 1 ;
```

Machine (PRG-SFC).Action Go_Up (ST)

```
0001  Y_pos := Y_pos - 1 ;
```

Machine (PRG-SFC).Aktion Count (ST)

```
0001  Counter := Counter + 1 ;
0002
0003
0004
0005
0006
```

modify_delay (FUN-ST)

```
0001  (* AMZ Feb 13 2012
0002
0003  Demo program to process screen buttons to modify alarm and warning
0004  timer values.
0005  *)
0006
0007  FUNCTION modify_delay : BOOL
0008  VAR_INPUT
0009    delay_number : WORD;  (* which delay to modify *)
0010    delay_change : INT; (* variance *)
0011  END_VAR
0012  VAR
0013    tempw : WORD;
0014    temp_delay : INT;
0015    temp_secs : WORD;
0016
0017  END_VAR
```

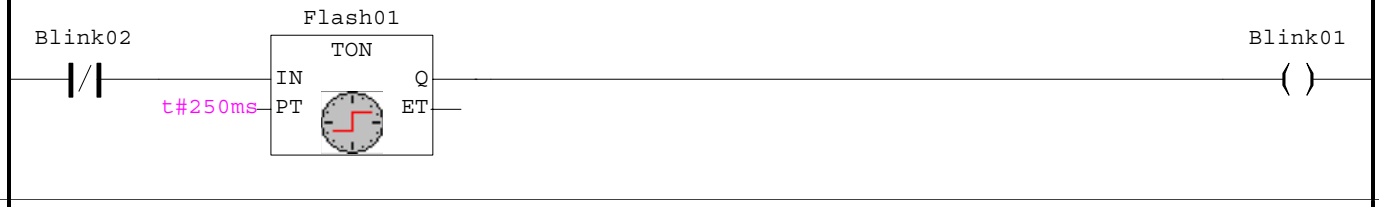
```
0001  (* change delay time in seconds based on passed incremental value *)
0002
0003  modify_delay := FALSE;
0004
0005  CASE delay_number OF
0006      1 : temp_delay := TIME_TO_INT(delay01);
0007      2 : temp_delay := TIME_TO_INT(delay02);
0008  END_CASE
0009
0010  temp_delay := temp_delay / 1000;  (* convert ms to seconds *)
0011  temp_delay := temp_delay + delay_change;  (* convert in whole seconds *)
0012
0013  IF temp_delay < 5 THEN temp_delay := 5; END_IF
0014  IF temp_delay > 99 THEN temp_delay := 99; END_IF
0015  temp_secs := temp_delay;  (* delay in seconds *)
0016
0017  temp_delay := temp_delay * 1000;  (* back to ms *)
0018
```

```
0019 CASE delay_number OF
0020   1 : delay01 := INT_TO_TIME(temp_delay); delay01_secs := temp_secs;
0021   2 : delay02 := INT_TO_TIME(temp_delay); delay02_secs := temp_secs;
0022 END_CASE
0023
0024 modify_delay := TRUE;  (* return valu *)
0025
```

Output_LED (PRG-LD)

```
0001 (* AMZ Feb 13 2012
0002
0003 Ladder logic to provide global flash timer.
0004 *)
0005
0006 PROGRAM Output_LED
0007 VAR
0008     Flash01: TON;
0009     Flash02: TON;
0010     Blink01: BOOL;
0011     Blink02: BOOL;
0012 END_VAR
```

0001



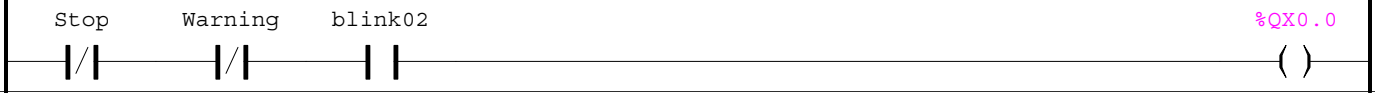
0002

Bool to control flashing

0003

0004

Normal run light

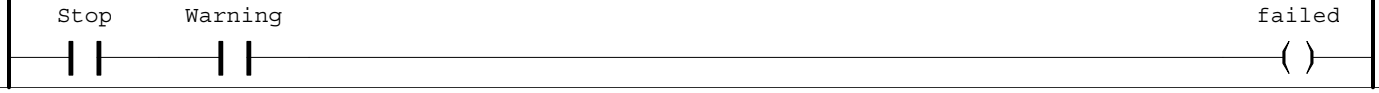


0005

Warning lightt

0006

Failed light

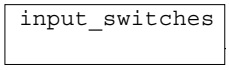


PLC_PRG (PRG-FBD)

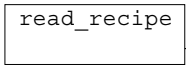
```
0001 (*  AMZ Feb 13 2012
0002
0003 Main POU for Wago PLC demonstration.
0004 This unit is called on program start and schedules all
0005 other units.
```

```
0006
0007 *)
0008
0009 PROGRAM PLC_PRG
0010 VAR
0011     Trig1: R_TRIG;
0012     Trig2: F_TRIG;
0013     Timer1: TOF;
0014     Timer2: TON;
0015     Timer3: TP;
0016     trig_observer: BOOL;
0017 END_VAR
```

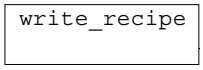
0001



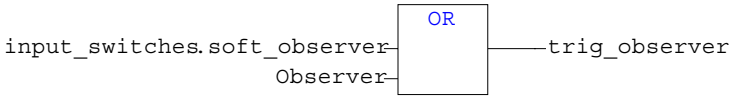
0002



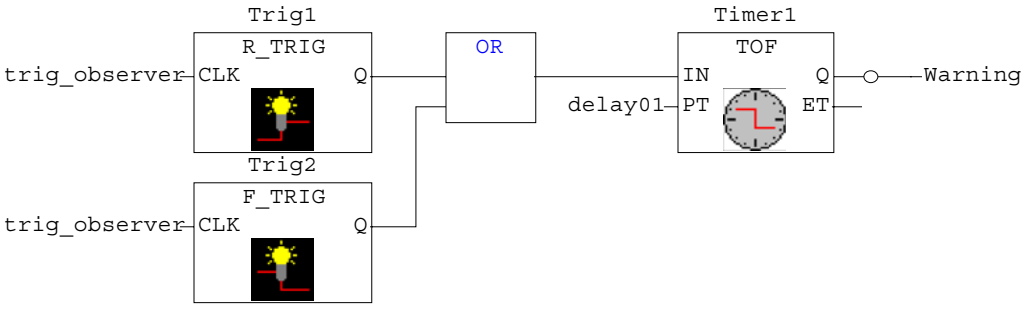
0003



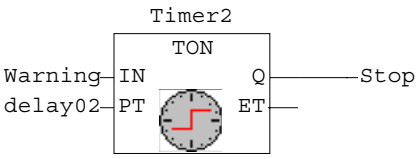
0004



0005

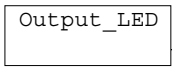


0006



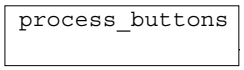
0007

```
Update the PLC indiator LEDs
```



0008

```
Proces the functions related to button presses
```



0009 do not continue if stopped

Stop Return

0010 Operate state machine when not stopped.

Machine

process_buttons (PRG-ST)

```
0001 PROGRAM process_buttons
0002 VAR
0003
0004 END_VAR
```

```
0001 (*  AMZ Feb 13 2012
0002     demo program button processor *)
0003
0004 (* increment passed value on first transition of the
0005 variable observer. The increment will depend on which linkable
0006 routine is present on the controller during runtime.
0007
0008 All versions of the demo function are called by the same
0009 function name MyTestFunction
0010 *)
0011
0012 (* A link to the following function won't resolve when
0013 project is downloaded to the target 750-880*)
0014
0015 IF observer   THEN
0016 (* demo_val := MyTestFunction(demo_val); *)
0017
0018   observer := FALSE;
0019 END_IF;
0020
0021
0022 (* process buttons for delay value change. *)
0023
0024 IF del01u THEN
0025   modify_delay(1, 1);  (* timer 1 up 1 *)
0026   del01u := FALSE;
0027 END_IF
0028
0029 IF del01d THEN
0030   modify_delay(1, -1);  (* timer 1 up 1 *)
0031   del01d := FALSE;
0032 END_IF
0033
0034 IF del02u THEN
0035   modify_delay(2, 1);  (* timer 1 up 1 *)
0036   del02u := FALSE;
0037 END_IF
0038
0039 IF del02d THEN
0040   modify_delay(2, -1);  (* timer 1 up 1 *)
0041   del02d := FALSE;
0042 END_IF
```

read_recipe (PRG-ST)

```
0001 PROGRAM read_recipe
0002 VAR
0003     fname : STRING := 'none';
0004     infile: DWORD;
0005     inbuffer: ARRAY[0..9] OF BYTE;
0006     readbytes: DWORD := 5;   (* read only 5 bytes, 2 values sep by comma or space *)
0007     file_read_count: DWORD;
0008     file_close_err : BOOL;
```

```
    tempw: WORD;
    new_recipe: INT;
END_VAR
VAR_OUTPUT
END_VAR
```

```
(* AMZ Feb 13 2012
Demonstration of Wago PLC program to read recipe information from a text file
using the file services on the PLC.

Four files can be read as determined by variable cur_recipe.

The text file has 5 signifiant characters. First 2 are ascii decimal of
warning delay.Then a space (or comma), then the stop delay.
Minimum delay is 5 seconds, max is 99.
*)

IF read_recipe_button THEN
      IF curr_recipe < 1 THEN
        new_recipe := 1;  (* try to load the first recipe file *)
      ELSE
        new_recipe := curr_recipe+1; (* next recipe file *)
      END_IF
     IF new_recipe > 4 THEN new_recipe := 1; END_IF  (* max of 4 recipes *)

    CASE new_recipe OF
        1: fname := rec_fname_01;
        2: fname := rec_fname_02;
        3: fname := rec_fname_03;
        4: fname := rec_fname_04;
        ELSE fname := rec_fname_01;
    END_CASE

    infile := SysFileOpen(fname, 'r');  (* try to open the file *)
    file_read_count := 0;  (* assume no bytes will be read *)
    IF (infile >0) THEN  (* file opend ok *)
        file_read_count := SysFileRead(infile, ADR(inbuffer), readbytes);
        file_close_err := SysFileClose(infile);
    ELSE
         load_def_recipe := TRUE;  (* set defaults if no file read *)
      END_IF

      IF (file_read_count >= readbytes) THEN    (* convert first 2 characters in the
                                                 array to delay0 word *)
(* remove ascii offset 48 *)
        tempw := ((inbuffer[0] - 48) * 10) + (inbuffer[1] - 48);
        IF tempw < 5 THEN tempw := 5; END_IF
        delay01_secs := tempw;
        tempw := tempw * 1000;  (* convert to milliseconds *)
        delay01 := WORD_TO_TIME(tempw);  (* warming timer *)
(* skip over separator at [2] and get next 2 digits *)
        tempw := ((inbuffer[3] -48) * 10) + (inbuffer[4] - 48);
        IF tempw < 5 THEN tempw := 5; END_IF
        delay02_secs := tempw;
        tempw := tempw * 1000;  (* convert to milliseconds *)
        delay02 := WORD_TO_TIME(tempw);  (* stop timer *)
        curr_recipe := New_recipe;
    END_IF

    read_recipe_button := FALSE;
END_IF

IF load_def_recipe THEN
    new_recipe := 0;  curr_recipe:= 0;  (* use defaults *)
    fname := 'None - using default';
    delay01 := default_delay; delay02 := default_delay;
    delay01_secs := TIME_TO_WORD(default_delay / 1000);  (* delay in seconds *)
    delay02_secs := delay01_secs;
    load_def_recipe := FALSE;
    read_recipe_button := FALSE;
END_IF
```

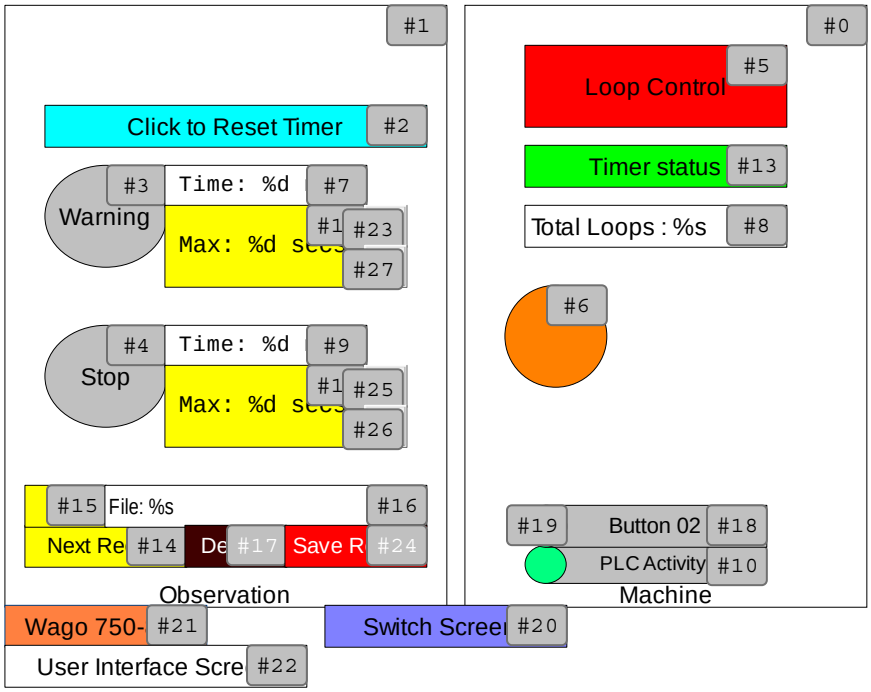
write_recipe (PRG-ST)

```
0001 PROGRAM write_recipe
0002 VAR
0003     fname : STRING := 'none';
0004     outfile: DWORD;
0005     outbuffer: ARRAY[0..9] OF BYTE;
0006     writebytes: DWORD := 4;   (* read only 4 bytes, 2 words *)
0007     file_write_count: DWORD;
0008     file_close_err : BOOL;
0009     tempw: WORD;
0010     save_recipe : INT;
0011     ii : WORD;
0012 END_VAR
0013 VAR_OUTPUT
0014 END_VAR
```

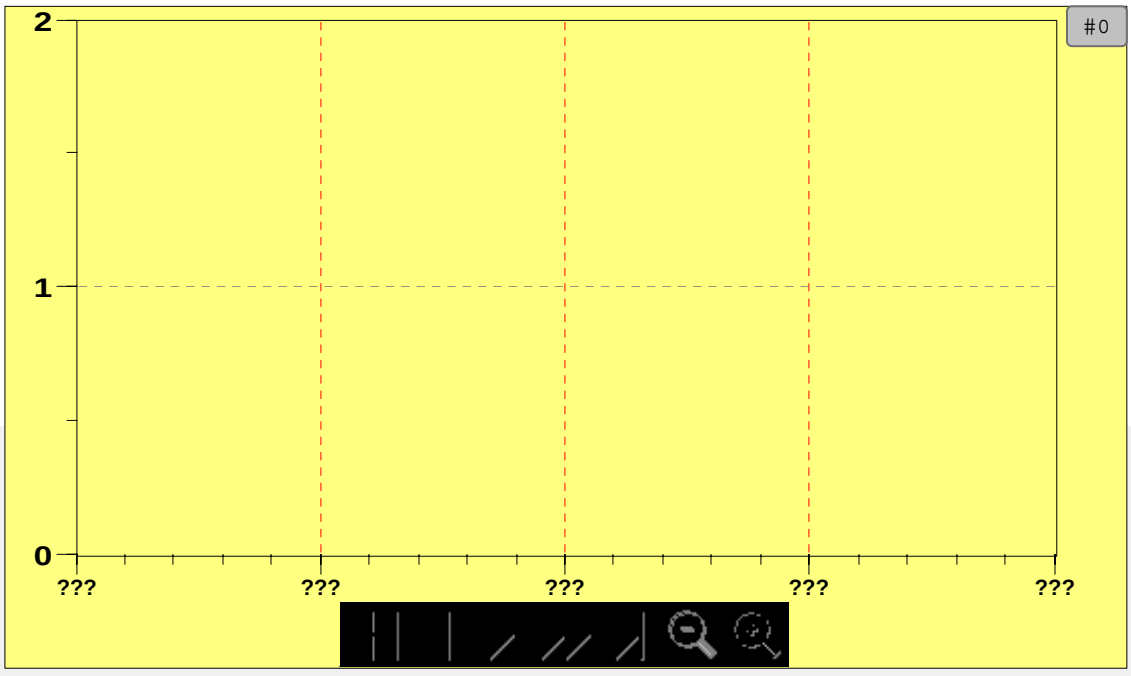
```
0001 (* AMZ Feb 13 2012
0002 demonstration of Wago PLC program to write date into a file using
0003 the file services on the PLC. The files to be written is determined by
0004 variable curr_recipe. The file has two 2-characer values in a simple
0005 text file in the format ABxCD.  AB is the 2 character representation
0006 of the warning delay in seconds. The middle character x can be anything,
0007 and a space is used in this example.
0008 The final pair CD is the 2 char representation of the alarm time in seconds.
0009 *)
0010
0011 IF write_recipe_button THEN
0012     save_recipe := curr_recipe; (* local of current global recipe *)
0013     IF ((save_recipe <1) OR (save_recipe > 4)) THEN
0014         load_def_recipe := TRUE;
0015         read_recipe; (* use default values *)
0016         write_recipe_button := FALSE;
0017     END_IF
0018 END_IF
0019
0020 IF write_recipe_button THEN  (* still OK to write *)
0021     save_recipe := curr_recipe;  (* max of 4 recipes *)
0022     CASE save_recipe OF
0023         1: fname := rec_fname_01;
0024         2: fname := rec_fname_02;
0025         3: fname := rec_fname_03;
0026         4: fname := rec_fname_04;
0027     END_CASE
0028
0029 (* put space characters into the array to delay0 word *)
0030     FOR ii := 0 TO 9 DO
0031       outbuffer[ii] := 32;  (* ascii space *)
0032     END_FOR
0033     tempw := TIME_TO_WORD(delay01) / 1000; (* milliseconds to secs *)
0034     IF tempw < 5 THEN tempw := 5; END_IF  (* min time allowed *)
0035     IF tempw > 99 THEN tempw := 99;  END_IF (* max delay secs *)
0036     IF tempw > 9 THEN  (* need 2 digits *)
0037       outbuffer[0] := WORD_TO_BYTE(tempw / 10) + 48;  (* upper digit *)
0038     ELSE
0039       outbuffer[0] := 48;  (* upper digit is ascii 0*)
0040     END_IF
0041     outbuffer[1] := WORD_TO_BYTE(tempw MOD 10) + 48;  (* lower *)
0042
0043     tempw := TIME_TO_WORD(delay02) / 1000;
0044     IF tempw > 99 THEN tempw := 99; END_IF
0045     IF tempw > 9 THEN
0046       outbuffer[3] := WORD_TO_BYTE(tempw / 10) + 48;  (* upper digit *)
0047     ELSE
0048       outbuffer[3] := 48;  (* upper digit is ascii 0*)
0049     END_IF
0050     outbuffer[4] := WORD_TO_BYTE(tempw MOD 10) + 48;  (* lower *)
0051     outbuffer[5] := 16#0A; outbuffer[6] := 16#0D;  (* cr/lf pair *)
0052
0053 (* write buffer to file *)
0054     outfile := SysFileOpen(fname, 'w');  (* try to open the file *)
```

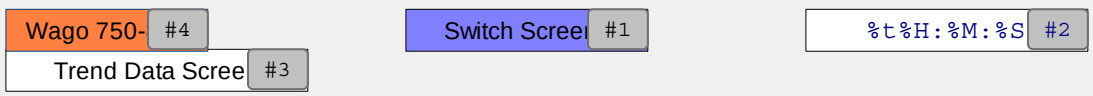
```
0055        writebytes := 10;  (* write all 10 bytes *)
0056        IF (outfile > 0) THEN (* file opend ok *)
0057            file_write_count := SysFileWrite(outfile, ADR(outbuffer), writebytes);
0058            file_close_err := SysFileClose(outfile);
0059        END_IF
0060        write_recipe_button := FALSE;
0061    END_IF
0062
```

PLC_VISU



TREND01





Data_points

```
0001
0002    VAR_GLOBAL
0003    END_VAR
0004
```

```
0005 VAR_GLOBAL CONSTANT
0006 END_VAR
0007
0008 VAR_GLOBAL RETAIN
0009 END_VAR
0010
0011 VAR_GLOBAL RETAIN
0012 END_VAR
```

Globale_Variablen

```
0001 VAR_GLOBAL
0002
0003     Observer : BOOL;
0004     Warning : BOOL;
0005     Stop : BOOL;
0006     Start : BOOL;
0007     flash_state : BOOL;
0008     demo_val : WORD;
0009   (*  count01 : BOOL;  *)
0010     default_delay : TIME := t#6000ms;
0011       delay01: TIME := t#6000ms;
0012     delay02: TIME := t#6000ms;
0013     delay01_secs, delay02_secs : WORD := 6; (* delay in whole seconds *)
0014
0015     failed: BOOL;
0016     read_recipe_button : BOOL;
0017     write_recipe_Button : BOOL;
0018     recipe_01 : BOOL;
0019     curr_recipe : INT :=0; (* current recipe in use, start with defaults *)
0020
0021 (* note: file names are platform specific *)
0022
0023     rec_fname_01 : STRING := 'recipe01.txt';
0024     rec_fname_02 : STRING := 'recipe02.txt';
0025     rec_fname_03 : STRING := 's:\RECIPE03.TXT';
0026     rec_fname_04 : STRING := 's:\RECIPE04.TXT';
0027
0028     load_def_recipe : BOOL := FALSE; (* this can be set globally *)
0029
0030     change_visu_button : BOOL := FALSE;
0031     soft_sel_visu : BOOL := FALSE;
0032     cur_visu: WORD := 1;
0033     new_visu: WORD := 1;
0034
0035     del01u : BOOL := FALSE;  (* button for delay01 up *)
0036     del01d : BOOL := FALSE; (* button for delay01 down *)
0037     del02u : BOOL := FALSE;  (* button for delay02 up *)
0038     del02d : BOOL := FALSE; (* button for delay02 down *)
0039 END_VAR
```

Input_Output

```
0001
0002 VAR_GLOBAL
0003 END_VAR
0004
0005 VAR_GLOBAL CONSTANT
0006 END_VAR
0007
0008 VAR_GLOBAL RETAIN
0009 END_VAR
```

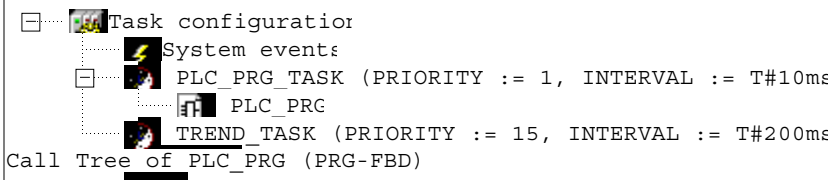
Variablen_Konfiguration

```
0001 VAR_CONFIG
0002 END_VAR
```

Globale_Variablen

```
0001 VAR_GLOBAL
0002 END_VAR
```

```
*PLC Configuration (Id.: 2010000001)
  Node number:  -1
  Input address: %IB0
  Output address: %QB0
  Diagnostic address: %MB0
  Download:  1
  AutoAdr:  1
Task configuration
```



Call Tree of PLC_PRG (PRG-FBD)